CRANFIELD UNIVERSITY

MAREN DAVID DANGUT

APPLICATION OF DATA ANALYTICS FOR PREDICTIVE MAINTENANCE IN AEROSPACE: AN APPROACH TO IMBALANCED LEARNING

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING
(Integrated Vehicle Health Management Centre)

Doctor of Philosophy (PhD)
Academic Year: 2018 - 2021

Supervisor: Professor Ian K. Jennions
Associate Supervisor: Dr Steve King
May 2021

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING
Integrated Vehicle Health Management Centre

Doctor of Philosophy

Academic Year 2018 - 2021

MAREN DAVID DANGUT

Application of Data Analytics for Predictive Maintenance in Aerospace: An Approach to Imbalanced Learning

Supervisor: Professor Ian K. Jennions
Associate Supervisor: Dr Steve King
May 2021

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

ABSTRACT

The use of aircraft operational logs to predict potential failure that may lead to disruption poses many challenges and has yet to be fully explored. These logs are captured during each flight and contain streamed data from various aircraft subsystems relating to status and warning indicators. They may, therefore, be regarded as complex multivariate time-series data. Given that aircraft are high-integrity assets, failures are extremely rare, and hence the distribution of relevant data containing prior indicators will be highly skewed to the normal (healthy) case. This will present a significant challenge in using data-driven techniques to 'learning' relationships/patterns that depict fault scenarios since the model will be biased to the heavily weighted no-fault outcomes.

This thesis aims to develop a predictive model for aircraft component failure utilising data from the aircraft central maintenance system (ACMS). The initial objective is to determine the suitability of the ACMS data for predictive maintenance modelling. An exploratory analysis of the data revealed several inherent irregularities, including an extreme data imbalance problem, irregular patterns and trends, class overlapping, and small class disjunct, all of which are significant drawbacks for traditional machine learning algorithms, resulting in low-performance models. Four novel advanced imbalanced classification techniques are developed to handle the identified data irregularities. The first algorithm focuses on pattern extraction and uses bootstrapping to oversample the minority class; the second algorithm employs the balanced calibrated hybrid ensemble technique to overcome class overlapping and small class disjunct; the third algorithm uses a derived loss function and new network architecture to handle extremely imbalanced ratios in deep neural networks; and finally, a deep reinforcement learning approach for imbalanced classification problems in log-based datasets is developed.

An ACMS dataset and its accompanying maintenance records were used to validate the proposed algorithms. The research's overall finding indicates that an advanced method for handling extremely imbalanced problems using the log-based ACMS datasets is viable for developing robust data-driven predictive maintenance models for aircraft component failure. When the four implementations were compared, deep reinforcement learning (DRL) strategies, specifically the proposed double deep State-action-reward-state-action with prioritised experience reply memory (DDSARSA+PER), outperformed other methods in terms of false-positive and false-negative rates for all the components considered. The validation result further suggests that the DDSARSA+PER model is capable of predicting around 90% of aircraft component replacements with a 0.005 false-negative rate in both A330 and A320 aircraft families studied in this research.

Keywords:

Extremely rare event, deep reinforcement learning, imbalance learning, predictive maintenance, aircraft

ACKNOWLEDGEMENTS

This work is dedicated to my family, who supported me throughout my studies.

I want to thank my supervisor, Prof. Ian K Jennions, who has always been a source of inspiration. I have been fortunate to have a supervisor who cared so much about my work and promptly responded to my questions and queries. His wisdom and supportive nature are second to none. I thank him for his consistent encouragement, especially when he assumes as my supervisor at my primary supervisor's resignation.

I would like to gratefully acknowledge the guidance, support, and encouragement of my associate advisor, Dr Steve King, for his continued mentorship, support and collaboration; at the early stage of the research, he logged many miles between Derby UK and Cranfield UK while he was visiting from the industry.

My gratitude extends to Dr Zakwan Skaf, my former supervisor. I would like to thank him for the guidance, encouragement and advice he has provided throughout my time as his student.

I will like to thank my colleagues from the IVHM centre. They provided me with incredible support throughout my PhD.

Finally, Special thanks to my wife and children for your continued support throughout my study.

This dissertation would not have been possible without my PhD funding from the Petroleum Technology Development Fund -PTDF of Nigeria.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

ACARS	Aircraft Communications Addressing and Reporting System
ACMS	Aircraft Condition Monitoring System
API	Application Programming Interface
ATA	The Air Transport Association
A330	A330 –Long-Range Aircraft Family
A320	A320 -Single-Aisle Aircraft
BITE	Built-in test Equipment
CBM	Condition-based Maintenance
CMS	Central Maintenance System
DL	Deep learning
DQN	Deep Q-Network
FDDP	Fault Detection, Diagnostics, and Prognostics
FIN	Aircraft Functional-Item Number
FWC	Flight Warning Computers
FDE	Flight Deck Effect
IR, ρ	Imbalanced Ratio
IoT	Internet of Things
LRU	Line Replacement Unit
MRO	Maintenance, Repair, and Operations
MPG	Mix Gaussian Process
ML	Machine learning
NLP	Natural Language Processing
OEM	Original Equipment Manufacturers
PdM	Predictive Maintenance
ROC	Receiver Operating Curve
SMOTE	Synthetic minority oversampling Techniques
SARSA	State Action Reward State Action
TF-IDF	Term Frequency–Inverse Document Frequency
1TX1	Air traffic service
4000HA	Pressure Regulating Valve
4001HA	High-Pressure Bleed Valve
438HC	Trim Air Valve
5RV1	Satellite Data unit
8HB	Flow control valve 2

10HQ	Avionics equipment ventilation computer
11HB	Flow control valve

CHAPTER 1: General Introduction

The concept of predictive and prescriptive maintenance in a complex system has recently gained more research attention, especially in the aviation industry. Before the advent of the internet of things (IoT) technologies, vehicle maintenance was mainly based on prearranged time-based schedules linked to the vehicle's age, the number of schedule cycles, or usage. It was not linked to the vehicle's real-time condition. Time-based maintenance is susceptible to unnecessary onsite vehicle inspections or visits to the service centre. Potential failures can go unnoticed between the schedules, and there are small or no useful insights for the airlines, OEM's and MRO's. The business cannot afford to let the asset run to failure, as it costs equipment damage and equipment downtime. The concept of prescriptive maintenance in a complex system tries to mitigate time-based maintenance challenges. Predictive maintenance is recently gaining more research attention, especially its application in the aviation industry [1]. The strength of predictive maintenance is in the ability to provide prognostics at the right time and correctly, which depends on the type of data used to train the models. Data imbalance distribution is a common challenge in many classification tasks. For example, when classifying a financial transaction, it is fraudulent or not. It is likely to have the majority of the transaction as legitimate and the minority as fraudulent. In this case, the training data contains highly imbalanced examples, reflecting the true class distribution in a general population.

Similarly, in aircraft' predictive maintenance, the imbalance of failure events to normal operation events can exist. This issue is due to the following reasons. The failure events usually rarely occur compared to the normal operation state for an in-service vehicle, and in total, there are fewer failure events.

The use of aircraft operational logs data to predict potential failure that may lead to disruption poses many challenges and has yet to be fully explored. These logs are captured during each flight and contain streamed data from various aircraft subsystems relating to status and warning indicators. They may, therefore, be regarded as complex multivariate time-series data. Given aircraft are high-integrity assets, failures are extremely rare, and hence the distribution of relevant data containing prior indicators will be highly skewed to the normal (healthy) case. This will present a significant challenge in using data-driven techniques to 'learning' relationships/patterns that depict fault scenarios since the model will be biased to the heavily weighted no-fault outcomes.

1.1 Motivation

This research work is motivated by the European AIRMES project [3], which focuses on optimising end-to-end aircraft maintenance activities in order to avoid operational disruptions. The research is also inspired by how it will impact aircraft maintenance by reducing operational disruptions, decreasing the average delay time, and improving aircraft utilisation through predictive and prescriptive maintenance. Prescriptive maintenance is advanced predictive maintenance. It leverages the approaches and capabilities of preventive, descriptive, and predictive maintenance to optimise system performance completely. Prescriptive maintenance is built on a technology that taps the power of IoT, big data analysis and machine learning to help vehicles become proactive participants in their maintenance [2]. This type of maintenance promises cost saving over time-based preventive maintenance because maintenance is carried out only when warranted. Preventive maintenance involves scheduling maintenance procedures in advance and performed periodically. However, when an unexpected failure occurs in-between the defined schedule period, the equipment becomes unavailable until this problem is fixed. Therefore, predictive maintenance aims to avoid such unexpected failures by continuous monitoring of the equipment condition and providing failure alerts well in advance to provide enough room to prepare for repairs. It gives a forecast of when equipment will fail so that maintenance can be systematically scheduled before the failure occurs.

1.2 Aim and Objectives

In order to address the research gaps mentioned above, the overall scientific aim and objectives are defined as follows:

Aim

This research aims to develop a data-driven predictive model for aerospace applications using advanced imbalanced classification algorithms.

Objective

In order to achieve the aim of the project, a series of objectives are set as follows:

1. To carry out a comprehensive literature review on the application of data analytics in aerospace, machine learning techniques, and then investigate approaches, and effects of imbalance problem in developing predictive modelling, also to understand the shortcomings and underpin the countermeasures to be designed.

2. To carry out data Pre-processing to quantify and understand various distribution and complexities inherent in the aircraft CMS datasets.

3. To design and develop a dynamic and robust Imbalance classification algorithm using machine learning, ensemble learning, deep learning and deep reinforcement learning (DRL) strategies that will handle extreme class imbalance, class overlapping and class disjunct in both binary and multi-class scenarios.

4. To develop an aircraft predictive maintenance model and test it using the different testing datasets to establish its adaptability to various challenges.

5. To validate the model using ground truth data in order to ascertain its accuracy and performance.

1.3 Research Methodology and Structure of the Thesis

This project consists of four (4) phases viz: 1. Comprehending context and literature review. 2. Data pre-processing, design, and development of the new algorithm. 3. building a predictive model. 4. Model validation, analysis of the result. 4. Final thesis write-up, as presented in Figure 1-1.

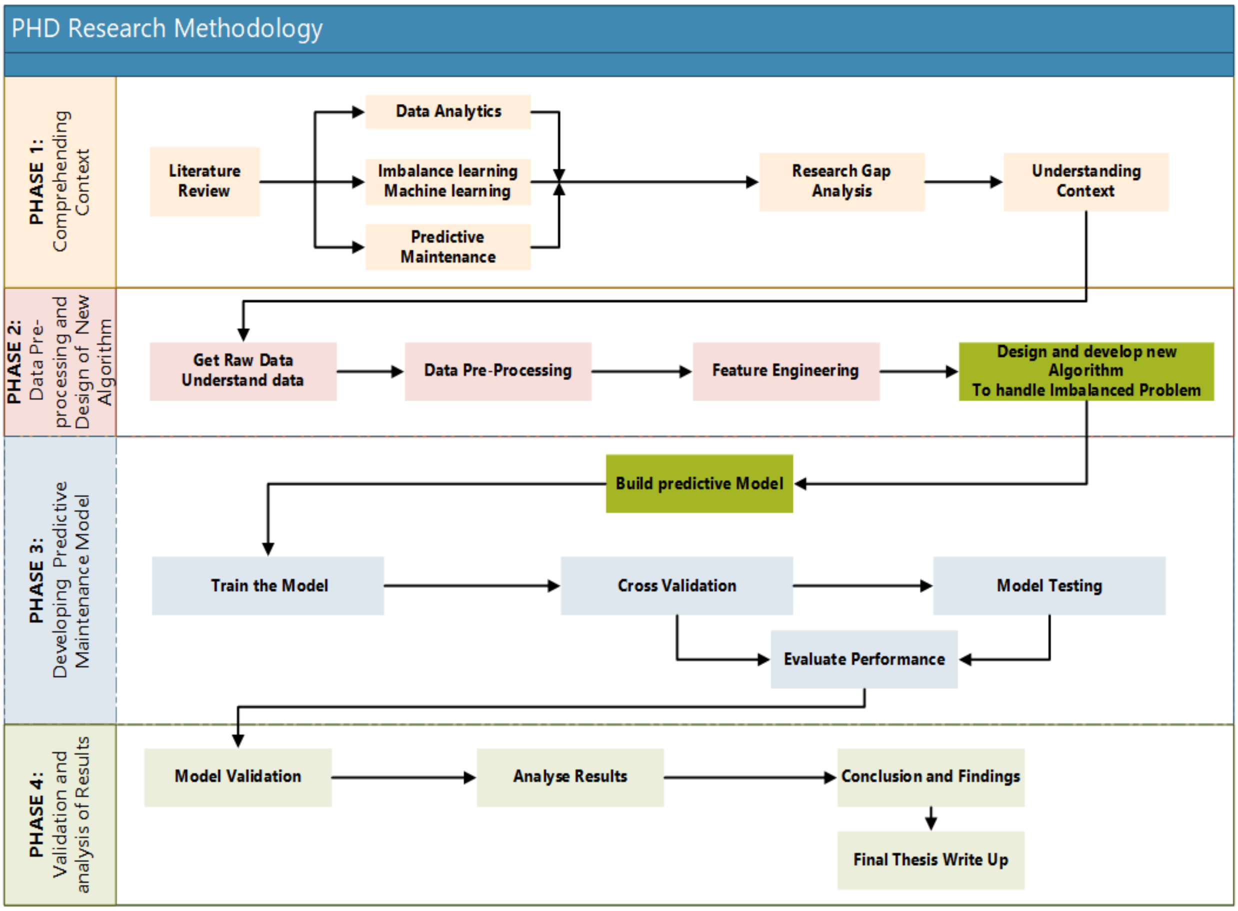


Figure 1-1 PhD Research Methodology Flow

Phase 1: In this phase, literature will be reviewed comprehensively on the application of data analytics and machine learning techniques in aerospace predictive maintenance. The effects of the imbalance problem and existing solutions to developing predictive modelling from aircraft log-based datasets will be investigated to understand the shortcomings and underpin the countermeasures to be designed. A review of methods of handling imbalance problems (both data, algorithm levels, and the hybrid) will be conducted. The review of literature generally will provide understanding and identification of various gaps existing in the knowledge. Finally, the general context of the subject will be understood.

Phase 2: In this phase, data pre-processing and feature engineering will be conducted on the aircraft operational log-based CMS datasets. After transforming the dataset for machine learning modelling, the data will be divided into two; 80% will be used for model training while 20% for model testing. Based on the findings in the literature and the result of an investigation carried out in phase one, the underpinned challenges will be formulated in a clear way for the new proposed algorithm development. Then the design and implementation of the novel algorithm will be carried out. The algorithm will focus on solving both binary and multi-class imbalance problems, tackling the extreme imbalance ratio, irregular patterns and trends and class disjunct in a big data context.

Phase 3: Based on the new algorithm developed in phase 2, a data-driven predictive model for conditioned based maintenance will be trained using the CMS training dataset. The new algorithm will iteratively be evaluated to meet the desired requirement. After training the algorithm, the resulting model will be tested using the testing dataset. The predictive model will be evaluated using matrics appropriate to imbalanced classification problems such as precision, recall, F1 score, G-mean and ROC curve. Instead of relying on the general accuracy, which is a bias towards the majority class.

Phase 4: In order to ensure the quality of our data-driven predictive model, validation will be carried out using ground truth data available and other related datasets. The ground truth data is an actual maintenance record carried out by aircraft maintenance engineers. Similarly, another dataset from a different type of aircraft will be used to ensure the predictive model's adaptability. Results obtained will be analysed, and conclusions will be made. Finally, a thesis write-up will be carried out.

1.4 Presentation of the Thesis

This thesis is organised as a series of chapters, each formatted as a paper for publication. There are Six technical chapters in all. All articles were written by the primary author, Maren David Dangut, edited and co-authored by Prof. Ian K. Jennions, Dr. Steve King, and Dr. Zakwan Skaf. Chapter 1 show a general thesis introduction. Chapters 2-7 are reformatted versions of published papers. Chapter 8 present a general discussion. The concluding remarks of this work and suggestions for future research are included in Chapter 9.

Chapter 1: The general research background and context are presented in this chapter.

Chapter 2. Exploratory data analysis is presented in chapter 2. The chapter shows data visualization, pre-processing, and feature engineering of the aircraft operational log-based CMS datasets.

Chapter 3: In this chapter, an overview of the related work is presented. This review is organised into three sections. The first section focuses on imbalanced learning techniques, and the second section provides the current trends on the application of data analytics in the aerospace industry. The third section gives a general overview of maintenance strategies in complex systems.

Chapter 4: This chapter describes how to deal with an extremely imbalanced log-based dataset using new data processing techniques. The raw log-based data has a variety of features, such as symbolic sequences, numeric time series, categorical variables, and unstructured text, which necessitates a thorough and meticulous processing strategy. The difficulty in predicting rare failure from a large log-based time-series dataset was determined to be due to the data distribution's irregular patterns and trends, which interferes with temporal feature learning. As a result, an algorithm was created to deal with the unique aspects of data pre-processing. The novel method combines natural language processing (NLP) with ensemble learning for pattern recognition and classification, transforming and integrating well-known NLP techniques (TF-IDF and Word vectorization) with ensemble learning. In terms of precision and recall, the suggested method performs around 10% better than the baseline method (Synthetic Minority Oversampling Technique- SMOTE). It was also discovered that the problem of class imbalance could be solved by focusing solely on patterns in the minority group. As a result, the classification performance of the model has improved. The accompanying paper contains documentation for the proposed implementation. ***(Paper Published - Dangut, Maren David, Zakwan Skaf, and Ian K. Jennions. "An integrated machine learning model for aircraft components rare failure prognostics with log-based dataset." ISA transactions 113 (2021): 127-139. DOI: 10.1016/j.isatra.2020.05.001*)**

Chapter 5: This chapter presents new proposed techniques for handling extreme imbalanced classification problems based on ensemble-hybrid learning using heterogeneous datasets. Two new algorithms are proposed and implemented: The

Balanced Calibrated Hybrid Ensemble Technique (BACHE) algorithm and the hybrid soft mixed Gaussian processes with the expectation-maximisation (EM) algorithm. The formulation of BACHE is based on the combination of hybrid-ensemble and cost-sensitive learning approaches to handle the extreme imbalanced problem. In the hybrid-ensemble algorithm, a balance-cascading algorithm is used to divide the data into subsets of the majority class. Then the minority class is synthesised and boosted using boosting data expansion policy, which overcomes the extreme imbalance classification problems and reduces the computational cost (efficient for larger datasets). Also, classifiers' unique arrangement and the introduction of cost sensitivity to each weak learner reduce variance and bias, producing improved performance. The BACHE algorithm shows to be robust and provides a high-performance solution for handling data imbalance problems - focusing on extreme imbalance ratio and irregular distribution (class drifting) in both binary and multi-class contexts. The implementation results showed that BACHE has a better performance than other similar ensemble and imbalance learning techniques. It also achieved a lower computational time; this makes it suitable for processing imbalanced datasets in a big data context. The approach also achieved a significant level of improvement in the reduction of false-positive and false-negative rates. The documentation of the BACHE is presented in a paper ***(Paper under review (minor correction): Dangut, Maren David, Zakwan Skaf, and Ian Jennions "Handling Imbalanced Data for Aircraft Predictive Maintenance using the BACHE Algorithm" Applied Soft Computing Journal).***

In addition, an enhanced method for dealing with imbalance classification problems in heterogeneous equipment datasets was developed. To improve the prediction of the minority class during learning, the technique uses a mixture of soft mixed Gaussian processes and the expectation-maximization (EM) algorithm. ***(Paper Published: Dangut, Maren David, Zakwan Skaf, and Ian Jennions. "Aircraft Predictive Maintenance Modeling using a Hybrid Imbalance Learning Approach." Available at SSRN 3718065 (2020)).***

Chapter 6: This chapter presents new proposed deep-learning techniques for handling extreme rare failure predictions. A novel loss function is derived for deep neural networks, enabling the deep learning algorithms to respond favourably to both minority and majority groups. The new approach presents a unique way of changing loss function with respect to weights and a unique arrangement of neural networks; it also dynamically regulates the combined weight to produce a merged predicting result. The approach was verified using LSTM networks. The LSTM model weights are combined at each time step adaptively and recursively by using past predictions' errors and discarded weight at the forget gate layer. This approach helps in addressing the class imbalance problem. The experiment result showed that the Rescaled-LSTM has a better performance than other similar imbalance learning techniques. A significant level of improvement in the reduction of false-positive and false-negative rates was achieved. The documentation of the proposed algorithm is published in the following paper ***(Paper Published: Dangut, Maren David, Zakwan Skaf, and Ian Jennions. "Rescaled-LSTM for Predicting Aircraft Component Replacement Under Imbalanced Dataset Constraint." 2020 Advances in Science and Engineering Technology International Conferences (ASET). IEEE. DOI: 10.1109/ASET48392.2020.9118253).***

Furthermore, the derived rescaled loss function was also used in the implementation of a proposed auto-encoder bidirectional gated recurrent network (AE-BGRU) model to predict rare failure. The result shows improved performance compared to a unidirectional approach. The documentation of the proposed algorithm is published in the following paper ***(Paper Presented at - Conference 4th IFAC Workshop on Advanced Maintenance Engineering, Services and Technologies- September 2020, Cambridge UK. Published - Dangut, Maren David, Zakwan Skaf, and Ian K. Jennions. "Rare Failure Prediction Using an Integrated Auto-encoder and Bidirectional Gated Recurrent Unit Network." IFAC-PapersOnLine 53.3 (2020): 276-282. DOI: 10.1016/j.ifacol.2020.11.045)***

The implementation of the AE-BGRU algorithm was extended to include more analysis and integration of convolutional neural network(CNN) into the model to improve predicting extreme failures in aircraft. The result is discussed and presented for Journal publication. ***(Paper Accepted: Maren David Dangut, Ian Jennions, Steve King and Zakwan Skaf "A Rare Failure Detection Model for Aircraft Predictive Maintenance Using Deep Hybrid Learning Approach" Neural Computing and Applications).***

Chapter 7: This chapter presents the implementation of deep reinforcement learning for the classification of an imbalanced dataset. In this approach, the problem is formulated as a Markov-decision process framework and solved using the deep reinforcement learning (e.g. deep Q-learning networks) techniques. As in reinforcement learning, the agent serves as a classifier, which performs classification action sequentially. The environment evaluates the classification action and returns a reward to the agent to make the next classification. A reward for the minority class is set higher, so the agent becomes sensitive to the minority class, which handles the extreme imbalance. Lastly, the agent finds the optimal classification policy. The interaction between state, action, and reward is stored in an experienced memory, and a mini-batch of the transactions is fitch and trained using deep neural networks. The documentation of the proposed algorithm is presented in the following paper ***(Paper under review (Minor correction): David Dangut, Ian Jennions, Steve King and Zakwan Skaf "A Rare Failure Detection Model for Aircraft Predictive Maintenance Using Deep Hybrid Learning Approach" Journal of Mechanical Systems and Signal Processing).***

Chapter 8: This chapter presents a general discussion of the result. The chapter focus on the findings of the research in terms of the real-world impact it will have.

Chapter 9: This chapter presents the general conclusions of the thesis and proposes areas for future research.

1.5 Reference

1. Khoshafian S., Rostetter C. Digital Prescriptive Maintenance: Disrupting Manufacturing Value Streams through Internet of Things, Big Data, and Dynamic Case Management. Pega Manufacturing. 2020; : 1–20. Available at: DOI:https://www.pega.com/system/files/resources/2019-01/Digital-Prescriptive-Maintenance.pdf

2. Setrag K., Rostetter C. Digital Prescriptive Maintenance. Internet of Things, Process of Everything, BPM Everywhere. 2015; : 1-20. Available at: DOI:https://www.pega.com/industries/manufacturing/digital-prescriptive-maintenance

3. Ferreira JF. AIRMES Newsletter 2020. 2020; (681858): 1–11. Available at: DOI:http://www.airmes-project.eu/files/newsletters/AIRMES_Newsletter4_September2019.pdf

Chapter 2: Exploratory Data Analysis for Aircraft Central Maintenance Dataset

This study was motivated by an increasing need for efficient and optimised data-driven machine learning approaches, notably for anticipating extremely rare events [1]. The research is based on the European AIRMES project [1], which is tasked with optimising end-to-end aircraft maintenance operations in order to avoid operating delays. One of their objectives is to develop a novelty identification system based on ACMS data and maintenance records. Thus, this project aims to develop robust algorithms for predictive models utilising the ACMS dataset. The ACMS data is valuable because it offers evidence of possible problems in aircraft operation and maintenance. The ACMS data analysis results can be used to develop a predictive model that can be used to improve aircraft operations and maintenance. The maintenance driver for this research, as aligned with AIRMES project, is to minimise overall maintenance and operating costs while increasing system uptime by using data-driven predictive modelling to mitigate unplanned or unscheduled maintenance in a fleet.

2.1. Description of ACMS Data and visualization

This study uses more than eight years' worth of data. The datasets are collected from two databases. The first database is the aircraft Central Maintenance System (ACMS) logs, which comprises error messages from BIT (built-in test) equipment (that is, aircraft fault report(s) record) and the flight deck effect (FDE). These messages are generated at different flight phases (take-off, cruise, and lading) stages. The second database is the corresponding records of aircraft maintenance activities. These databases are associated with a fleet of civil aircraft. Usually, in aircraft, the primary purpose of ACMS is to facilitate maintenance activities by directly alerting fault messages. Pilots and maintenance engineers can use that; to at the main base-perform troubleshooting or at the line stop level -perfume component removal [2]. The primary function of aircraft CMS is to acquire and store messages transmitted by the connected system Built-In Test Equipment (BITE) or by Flight Warning Computers (FWCs), as seen in Figure 2-1.

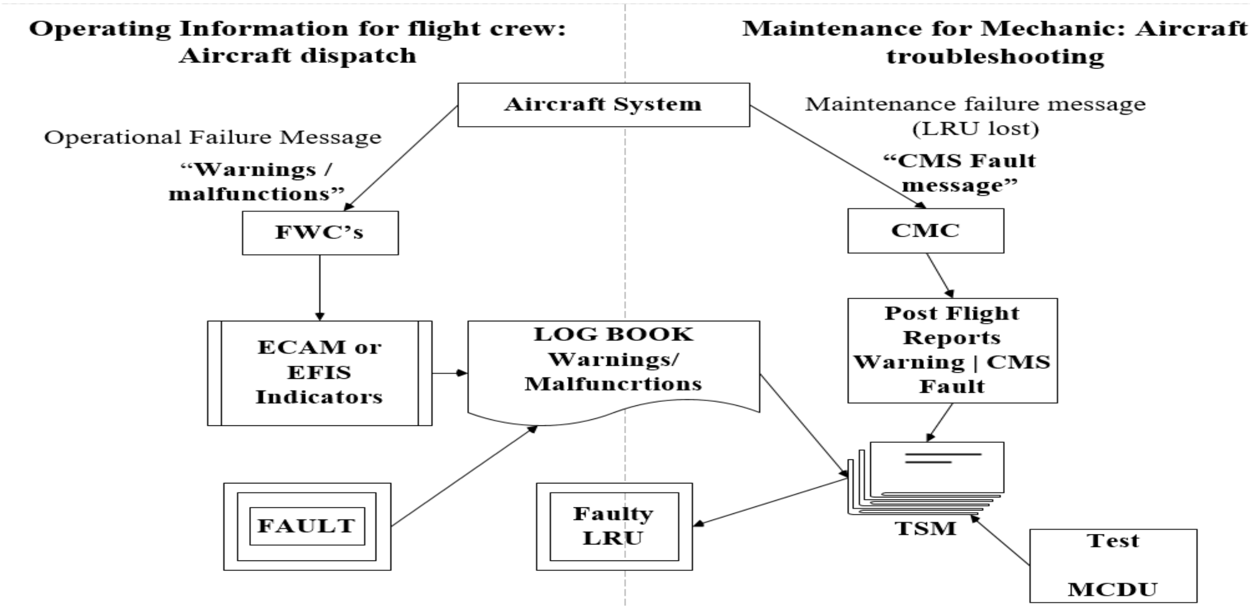
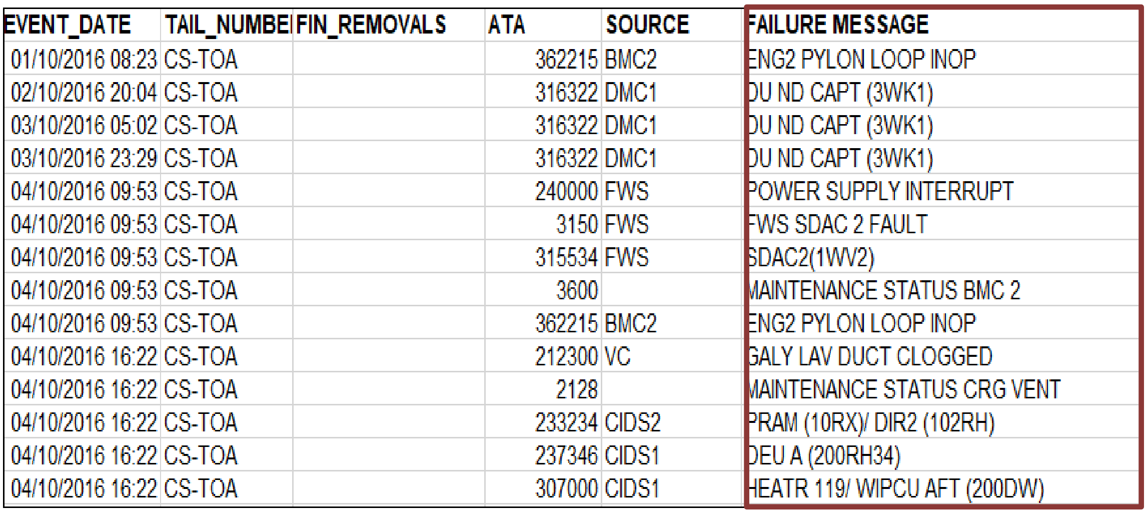

Operating Information for flight crew: Aircraft dispatch
Maintenance for Mechanic: Aircraft troubleshooting
Aircraft System
Operational Failure Message
"Warnings / malfunctions"
Maintenance failure message (LRU lost)
"CMS Fault message"
FWC's
CMC
ECAM or EFIS Indicators
LOG BOOK Warnings/ Malfuncrtions
Post Flight Reports Warning | CMS Fault
FAULT
Faulty LRU
TSM
Test
MCDU

Figure 2- 1 Traditional Troubleshooting Philosophy in A330 CMS [2]

Sensors and monitoring systems are typically installed and configured in aircraft to monitor various components. Failure messages are created based on the configured rules when any configured rules are broken. According to Airbus training materials [48], each time a fault is detected and isolated, a failure message is generated by system BITE. The message is memorized in the BITE memory and transmitted to the CMS. Each failure message is made up of 48 characters long, composed of a faulty line replaceable unit (which is made up of one or more parts depending on the type) and an ATA 6-digit reference number. A message might contain several Line Replacement Units (LRU), but only one suspected element is faulty. Each message syntax is of the form B-FIN-BUSNAME; B (Most probable suspected component) – FIN (Functional Item Number) – BUS NAME (complementary information) as seen in Figure 2- 2



EVENT_DATE	TAIL_NUMBE	FIN_REMOVALS	ATA	SOURCE	FAILURE MESSAGE
01/10/2016 08:23	CS-TOA		362215	BMC2	ENG2 PYLON LOOP INOP
02/10/2016 20:04	CS-TOA		316322	DMC1	DU ND CAPT (3WK1)
03/10/2016 05:02	CS-TOA		316322	DMC1	DU ND CAPT (3WK1)
03/10/2016 23:29	CS-TOA		316322	DMC1	DU ND CAPT (3WK1)
04/10/2016 09:53	CS-TOA		240000	FWS	POWER SUPPLY INTERRUPT
04/10/2016 09:53	CS-TOA		3150	FWS	FWS SDAC 2 FAULT
04/10/2016 09:53	CS-TOA		315534	FWS	SDAC2(1WV2)
04/10/2016 09:53	CS-TOA		3600		MAINTENANCE STATUS BMC 2
04/10/2016 09:53	CS-TOA		362215	BMC2	ENG2 PYLON LOOP INOP
04/10/2016 16:22	CS-TOA		212300	VC	GALY LAV DUCT CLOGGED
04/10/2016 16:22	CS-TOA		2128		MAINTENANCE STATUS CRG VENT
04/10/2016 16:22	CS-TOA		233234	CIDS2	PRAM (10RX)/ DIR2 (102RH)
04/10/2016 16:22	CS-TOA		237346	CIDS1	DEU A (200RH34)
04/10/2016 16:22	CS-TOA		307000	CIDS1	HEATR 119/ WIPCU AFT (200DW)

Figure 2- 2 An example of a real CMS message with event date, aircraft tail number, LRU, ATA reference number, and maintenance message

All the CMS failure messages are recorded in a logbook. Unplanned maintenance can be scheduled for malfunctioned items based on the failure messages. After the maintenance, the engineers update maintenance records with the repair details. The maintenance record provides detailed information about each component or item replaced (such as the repair date, part identification number, time spent on troubleshooting, etc.).

The ACMS log data is obtained from a fleet of civil aircraft. The data is distinctive in a variety of ways. It can be seen as a numeric time-series or symbolic sequence with features extracted from failure message or event occurrences over some segments or window periods. It contains categorical values, both text and numerical, as in the case of failure source, failure type, and ATA number. The usual way to predictive maintenance with this type of data is to analyse historical failure messages for irregularities using domain experts with experience. Then, using some preconditioned criteria, predictive patterns can be manually formed for a specific component based on this observation. This is a very specialised and time-consuming strategy that requires considerable expertise and experience. On the other hand, the traditional approach establishes a key concept: system failure can be predicted by evaluating its failure history. This study is motivated by the traditional concept and serves as the foundation for our problem formulation. ACMS data has been used mostly for short-term troubleshooting, anomaly identification, LRU removal, and system failure study or testing. There has been little research on the use of this type of data to develop predictive maintenance models.

The dataset is acquired from a fleet that consists of two distinct aircraft families: long-range (A330) and short-aisle (A320); the data is classified accordingly. The aircraft classification is important since the data generated varies in terms of attributes and structure depending on the type of aircraft. Other distinguishing aspects are the route designations; some were reserved for long-distance routes, while others were reserved for short-distance routes. Components replaced owing to unplanned maintenance are targeted in each family, and their failure behaviours are investigated. The behavioural patterns are then used to build a predictive model to predict their replacement, and the behavioural patterns are referred to as the failure frequency distribution across the fleet for each type of fault. Unscheduled replacement is considered because of the high cost of maintenance associated with unplanned failures. Airlines and MRO's have to cut costs wherever possible to participate in a market under excessive cost pressure. A predictive model for unscheduled maintenance events could help to avoid expensive excesses. Therefore, developing a predictive model to predict the upcoming unplanned failure (early warning of systematic failures of aircraft components) can reduce the overall maintenance and operation cost.

The ACMS data requires a thorough study in collaboration with a domain expert to find links between qualities and the "decision" variables of interest and cause-effect relationships for failure. Because a description does not accompany the ACMS data, the initial objective was to comprehend the data characteristics and then determine its suitability for predictive modelling.

Columns (variables) available in the ACMS dataset are :

1. Event date: Date the failure message occurs

2. Aircraft Tail Number: Uniquely identified aircraft in the fleet data

3. FIN Removals: Identify the components removals

4. Failure Source: This Shows the subsystem that the failure message belongs to

5. Failure Message: Show the description of the failure message (Each message syntax is of the form B-FIN-BUSNAME; B (Most probable suspected component) – FIN (Functional Item Number) – BUS NAME (complementary information))

6. Leg of occurrence: Indicates the flight where the failure message occurs

7. TSI(FH): It shows time Since the installation of the component replaced

8. CSI (FC) : Cycles Since Installation (cycles)

9. Date Install (DT_INST): This shows the date the component/LRU was installed

10. DT_REM: Component/LRU Removal Date

11. RAZAO_REMO: Reason for Removal either as Scheduled or Unscheduled.

12. SIT: Situation at Removal either Serviceable or Unserviceable

13. Flight Phase: This shows the exact flight phase when the failure message was generated (e.g. take-off, cruise, and landing)

14. Departure Airport: Show the take-off airport

15. Arrival Airport: Destination Airport

16. Flight Number: show the flight number assign to the particular aircraft

2.2 Maintenance Records Data Visualization

The second database contains information about aircraft maintenance activities covering the period in question. The maintenance record contains information about the components or LRUs replacements. The features relevant for labelling the ACMS dataset are the removal date, the reason for the removal (scheduled, unplanned, or for convenience), and the aircraft tail number.

The fist analysis in on the fleet of A330 family. The following LRU replacements are available from the A330 aircraft designated by their functional identification numbers (FINs): 8XS, 8KA, 705QN, 700QN1, 601QL1, 601QL2, 702QN, 5HA1, 5202GG, 703QN, 1WT1, 4000HA, 5RV1, 1SQ1, 59KV20, 4001HA, 8GV1, 3FP1, 3FP2, 2CE3, 5100KB, 59KH18, 3FP3, 4104KS1, 59KA10, 5426GG, 59KH23, 516KB, 4113KS, 1WT3, 1WT2, 7150HA1, 5RV1, 438HC, 1SH1, 1SH2, 701QN, 4000KS, 4019KS, 1SG, 1JG, 3GV1, 11GV2, 10GV1, 10GV2, 19FP1, 19FP3, 19FP8, 1SA2, 12HA1, 4GZ, 7HA1, 5801GG, 5403GG, 709QN, 5151JM2, 1KS1, 5404GG, 5427GG, 4506KS, 603QL2, 603QL1, 1SA1, 7GV1, 5GV2, 12GV1, 6GV2, 4000JG2, 1SQ2, 2CE1, 5QM1, 7SG1, 7SG2, 1HA1, 2GV, 5GV1, 19FP5, 19FP7, 5GA1, 1WW1, 1WW2, 1TX1, 500QU2, 5500QA1, 5QM2, 5491GG, 4057KS, 4511KS, 5208QS, 500QU1, 6GV1, 8GV2, 12GV2, 4GV2, 59KD, 4GV1 , 280HN, 9GV1,, 9GV2, 11GV1, 1HA2, 2GK, 59KE21, 13HA1, 5009EN, 5490GG, 5048EG2, 7GV2, 19FP4, 4010EG2, 19FP6, 16RV, 19RV1, 5216QS, 19FP2, 4112KS, 9KS1, 4010EG1 5048EG1.

Within the time under consideration, a total of 2062 LRU replacements were documented in the fleet of A330 maintenance records, including 1124 unscheduled replacements, 183 scheduled replacements, and 775 convenience replacements. The LRU removal is denoted in Figure 2-3 by their functional identification numbers. As seen, some replacements are more frequent than others.

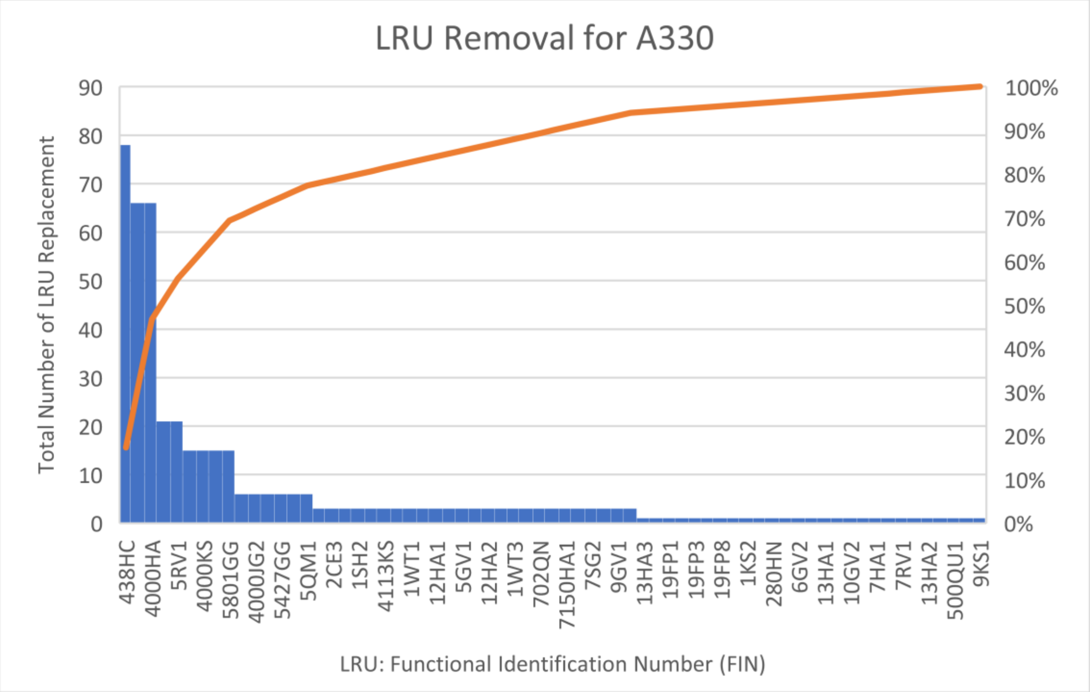


Figure 2- 3 The number of LRU removals associated with A330 Aircraft

According to the airline from whom this data was gathered, certain unscheduled component replacements have a greater impact on business than others, as illustrated in Figure 2-4. The higher the frequrncy means more replacement occur which increases maintenance cost.

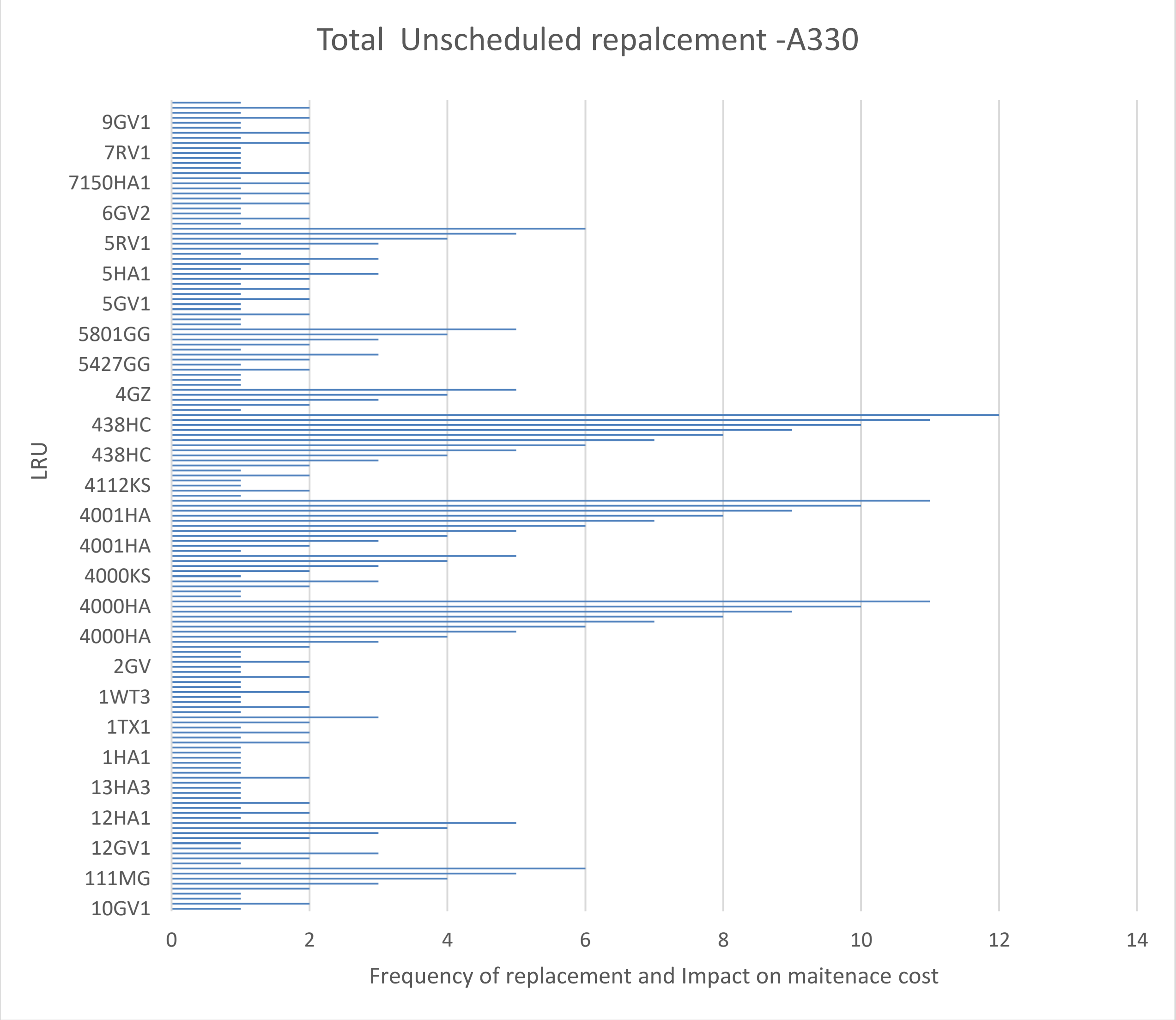


Figure 2- 4 Impact of Unschedule LRU Removals related to A330 Aircraft

Table 2-1 summarises the selected components related to A330 aircraft. The selection is based on their impact on maintenance cost, and the availability of minimum required patterns for training machine learning algorithms. The selected components are used in this study to validate the proposed machine learning algorithms.

The training data was divided into 80% training and 20% testing. The train/test data ranges from 1 January 2011 to 30 September 2016. The data was divided into 80% for training and 20 % for testing datasets. The validation dataset ranges from 30 September 2016 to April 2018.

Table 2- 1 Summary of Selected components removals associated with A330

Fleet of A330				
	Total (train and test)	Scheduled	Unscheduled	Unspecified
4000KS	151	80	53	18
4000HA	137	55	71	11
5RV1	62	15	28	19
438HC	46	15	18	13

The second analysis in on the fleet of A320 family. The following LRU replacements are related to the A320 aircraft, designated by their functional identification numbers (FINs): 11HB, 10HM3, 11HM3, 27HH, 7HH, 3CC2, 10WQ, 19FP1, 19FP2, 19FP3, 1KS1, 1KS2, 4001HA, 8XS, 15HQ, 8HB, 18HQ, 10HQ, 1CC1, 5WH, 2WH, 1WH, 6WH, 4WH, 3WH, 4CC, 3CC1, 3FP2, 3FP1, 3FP3, 10HA2, 5HA2, 4000HA, 24HQ, 19FP4, 19FP8, 22HQ, 1CC2, 1FP1, 1FP2, 5HA1, 22FN, 10HA1, 1HA2, 1HA1, 4005KM, 4015KM, 10CC, 1TX1, 30HH, 8022KM, 1WW1, 1WW2, 1TW, 59KD, 10WH, 16HQ, 1WV1, 1WV2, 1WT1, 9WH, 7WH, 10HH, 4QC, 1WD, 1WT2, 57HH, 23HQ, 8WH, 23HB, 47HH, 24HB, 19FP7, 20LP.

Within the period under consideration (training and testing), a total of 3239 LRU replacement was recorded, of which 1311 were Unscheduled, 1067 Scheduled and 861 Convenience. Figure 5-3 shows the frequency of LRU removals for the A320 aircraft family.

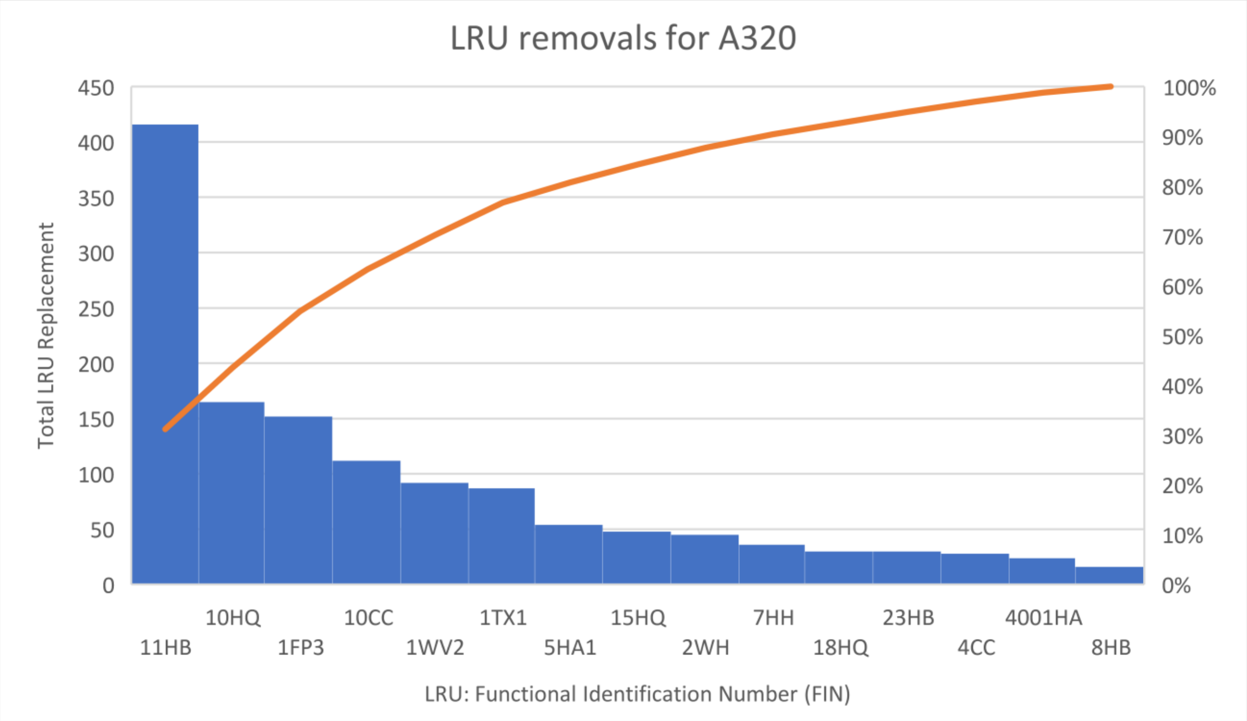


Figure 2- 5 The number of LRU removals associated with A320 Aircraft

Table 2-2 shows the summary of selected FIN from A320 aircraft (used to label the corresponding ACMS dataset): Dataset ranges from 1 January 2011 – 30 September 2016. It was divided into training (80%) and testing datasets (20%). The validation dataset ranges from 30 September 2016 - April 2018.

Table 2- 2 Summary of Selected components removals associated with A320

FIN	Total (Training &Testing)	Scheduled	Unscheduled	Unspacified
11HB	245	121	114	10
10HQ	120	78	21	21
1TX1	80	44	26	10
8HB	207	104	95	8

Among the available components/LRUs, this thesis concentrated on those with the greatest economic impact during operation, as determined by the airline that contributed the dataset, as shown in Table 2-3.

Table 2- 3 Selected Components to be considered in this study

Description of the Selected Components	
A330 Aircraft Family	A320 Aircraft Family
4000KS – Electronic Control Unit/ Electronic Engine Unit	**11HB** - Flow control valve
4001HA/4000HA – High-Pressure Bleed Valve	**10HQ** - Avionics equipment ventilation computer
5RV1- Satellite Data unit	**1TX1** - Air traffic service
438HC -- Trim Air Valve	**8HB** - Flow control valve 2

According to a review of maintenance records associated with the ACMS aircraft family, the total number of failure/warning messages for the A330 family after pre-processing is approximately 389902 in 4023 flights. The A320 family contains approximately 890120 in 10874 flights. According to the data, unscheduled component replacement occurs approximately two to three times every thousand flights.

2.3 Data Preprocessing and Transformation

The two critical processes in developing machine learning models are data pre-processing and feature engineering. Pre-processing is the process of cleaning data, whereas feature engineering is about developing new features. The following procedures were followed to engineer and pre-process features using the ACMS dataset.

Step 1- Handling Data Errors: Some data errors were observed in the raw dataset (A330 and A320), such as missing values, incorrect time and date associated with failure events. Other attributes such as flight Legs, departure and arrival airports provided additional information about the data's date and time for events failure. Therefore, a method known as imputation using other features [3] is used to handle missing data. The other option could be to drop the null, but dropping the null can reduce the data as more data is needed to train the machine learning models. Also, it can further worsen the problem because of the extreme imbalanced nature of the dataset. With the help of a domain expert, some failure (text) messages starting with some keywords are unnecessary. Thus, string comparison techniques were used to remove unwanted failure messages.

Step 2 -Labelling ACMS Data Using the Maintenance Records. The maintenance record contains information about the component's replacement. In labelling the ACMS data, the removal date and tail number from the maintenance record are used in correspondence to the event data and tail number in the ACMS. This information gives an idea about the last flight leg before removal. The leg that component replacement occur is labelled as failure leg (Positive), whereas others are labelled as non—failure leg (Negative).

Step 3 - Selecting the Right Features for Predictive Modelling.

Feature Importance Analysis: In selecting the best feature for modelling, feature importance provides insight into the dataset; the score highlight which features are more and less relevant to the target variable. Knowing feature importance can give meaningful information, such as determining a need to gather more data or using different data for predictive modelling [4]. In this study, the importance of the features was determined with the help of analysis and domain experts. The domain expert provided information about which variables are mostly used for troubleshooting failure related to each component, while the analysis reveals other hidden correlations between observed and latent variables.

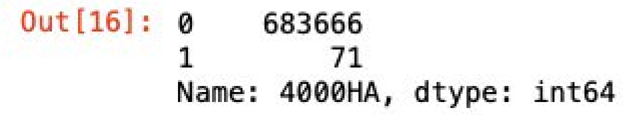
The first step is to split the dataset into dependant and independent features and select the right independent variables which will influence the dependent variable. The aim is to develop a model that will predict failure or each component in aircraft using the ACMS dataset. Each failure or replacement component is determined by a history of failure or warning messages in the dataset. Therefore, each target component is represented as a dependent variable (eg '4001HA', '4000KS', '438HC', '5RV1', '4000HA') while the related failure messages represent the independent variables (eg 'ATA', 'SOURCE', 'FAILURE MESSAGE',).

The following features were selected:

A330 Feature Index (['EVENT_DATE', 'TAIL_NUMBER', 'FIN_REMOVALS', 'ATA', 'SOURCE', 'FAILURE MESSAGE', 'LEG_OF_OCCURRENCE', 'TSI(FH)', 'CSI(FC)', 'DT_INST', 'FLIGHT_PHASE', 'DEPARTURE_AIRPORT', 'ARRIVAL_AIRPORT', 'FLIGHT_NUMBER', 'LEG_ID', '4001HA', '4000KS', '438HC', '5RV1', '4000HA'], dtype='object')

A320 Feature Index (['EVENT_DATE', 'TAIL_NUMBER', 'FIN', 'ATA', 'SOURCE', 'FAILURE MESSAGE', 'LEG_OF_OCCURRENCE', 'TSI(FH)', 'CSI(FC)', 'DT_INST', 'FLIGHT_PHASE', 'DEPARTURE_AIRPORT', 'ARRIVAL_AIRPORT', 'FLIGHT_NUMBER', 'LEG_ID', 'AIR_PACKxFAULT', 'VENT_AVNCS SYS_FAULT', 'DATALINK_ATSU_FAULT', '11HB', '10HQ', '1TX1', '8HB'], dtype='object')

The target or dependent variables are: '4001HA', '4000KS', '438HC', '5RV1', '4000HA', '11HB', '10HQ', '1TX1', '8HB'. The patterns related to dependent variable can either be 1 or 0, if the value is 1 it indicate component failure while 0 indicate non-component failure. For example for 4000HA as displayed below



```
Out[16]: 0    683666
         1        71
         Name: 4000HA, dtype: int64
```

It shows 683666 warning messages resulting in 71 failure/replacements

To visualise the link between dependent and independent features, a correlation matrix with a heatmap is plotted. As illustrated in Figures 2- 6 and 2-7 for the A330 and A320, correlation maps visually indicate the relationship between variables. Any independent variables that have a strong correlation (completely correlated features) with the dependent variables are undesirable because they introduce multicollinearity into the model, which reduces its predictive accuracy. As seen in A330 and A320, all independent variables have a correlation coefficient less than 0.5 with all dependent variables, which is desirable.

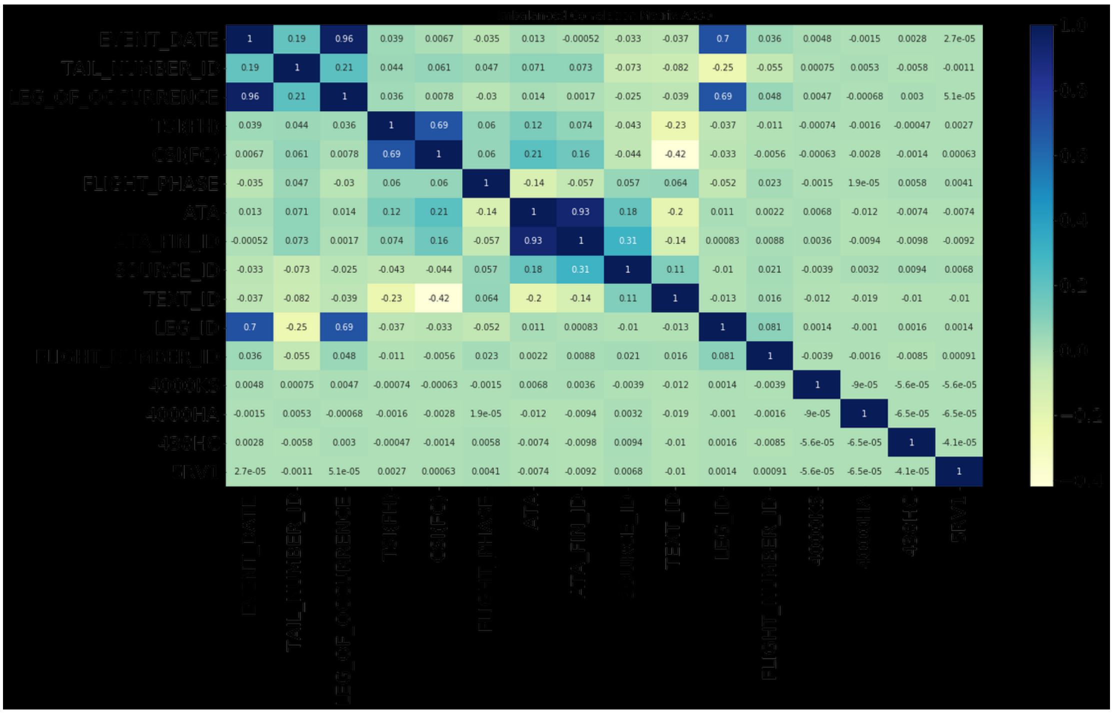


Figure 2- 6 Correlation Heatmap for A330 dataset

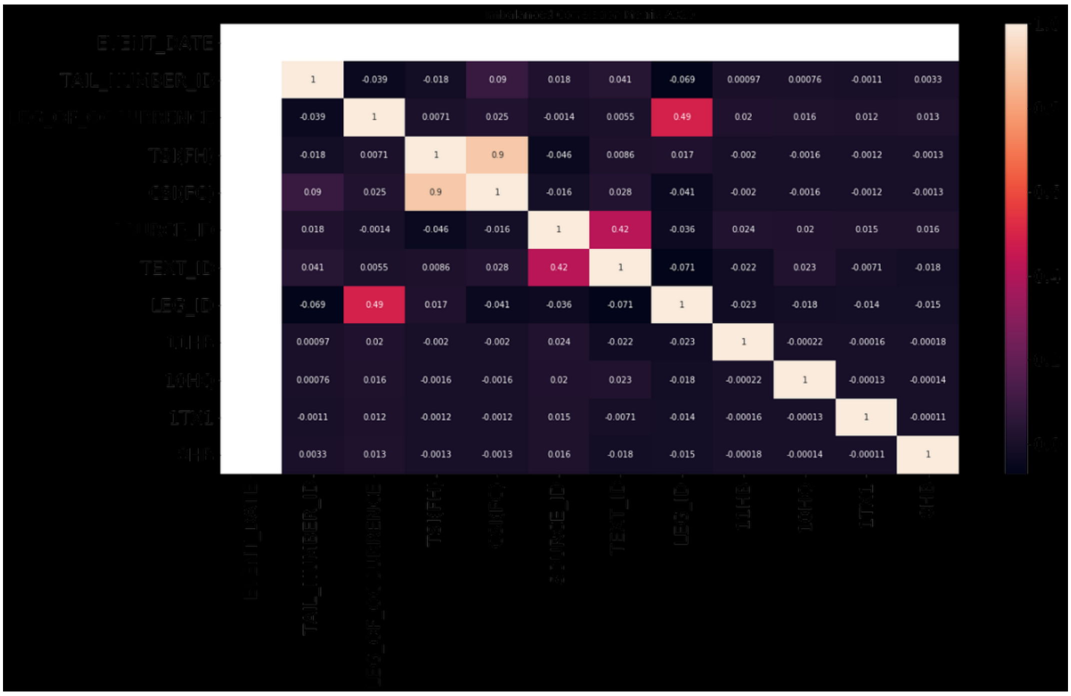
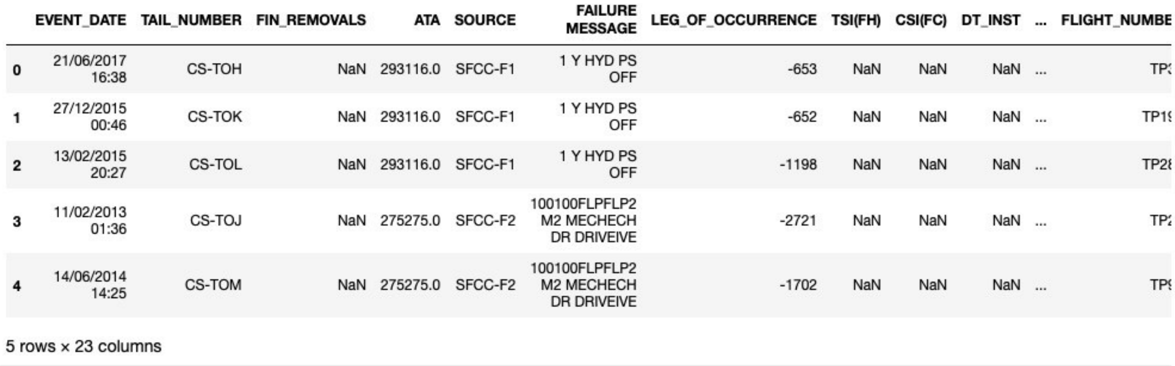


Figure 2- 7 Correlation Heatmap for A320 dataset

Step 4 -Creating New Feature:

In the ACMS data, some features are numerical while others are textual and categorical. Those features were all transformed to numerical for machine learning modelling. In the process of transforming the data, new features from the existing variables are created to improve the quality of the predictive model. The features were created using integer encoding and the one-hot encoding methods. The choice of the feature conversion process is based on the nature of the dataset because the data is heterogeneous with categorical features, which is not suitable for training machine learning algorithms in their original form. Figure 2-8 shows a section of the original ACMS data.

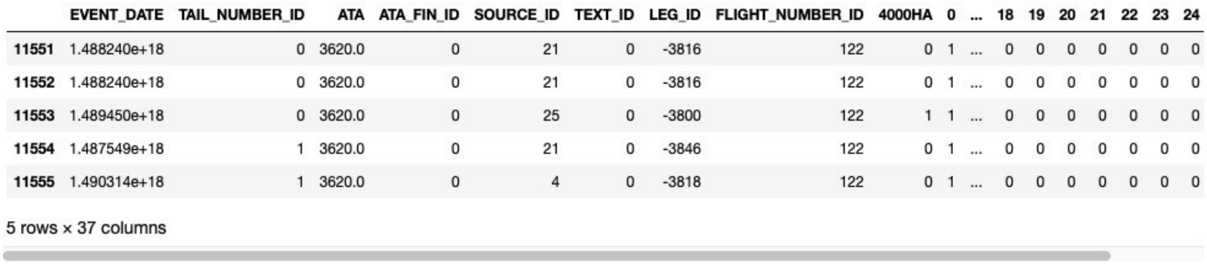


	EVENT_DATE	TAIL_NUMBER	FIN_REMOVALS	ATA	SOURCE	FAILURE MESSAGE	LEG_OF_OCCURRENCE	TSI(FH)	CSI(FC)	DT_INST	...	FLIGHT_NUMBE
0	21/06/2017 16:38	CS-TOH	NaN	293116.0	SFCC-F1	1 Y HYD PS OFF	-653	NaN	NaN	NaN	...	TP:
1	27/12/2015 00:46	CS-TOK	NaN	293116.0	SFCC-F1	1 Y HYD PS OFF	-652	NaN	NaN	NaN	...	TP1!
2	13/02/2015 20:27	CS-TOL	NaN	293116.0	SFCC-F1	1 Y HYD PS OFF	-1198	NaN	NaN	NaN	...	TP2
3	11/02/2013 01:36	CS-TOJ	NaN	275275.0	SFCC-F2	100100FLPFLP2 M2 MECHECH DR DRIVEIVE	-2721	NaN	NaN	NaN	...	TP:
4	14/06/2014 14:25	CS-TOM	NaN	275275.0	SFCC-F2	100100FLPFLP2 M2 MECHECH DR DRIVEIVE	-1702	NaN	NaN	NaN	...	TP!

5 rows × 23 columns

Figure 2- 8 Section of the original ACMS data

First, the input variables were converted into numerical to allow easy learning by machine learning algorithms. Then for variables where ordinal relationships exist, an integer encoding was used, and where such a relationship does not exist, one-hot encoding was used. Failure messages, for example, are categorical, so One-hot encordin is employed.. Figure 2-9 shows a section of the ACMS data after transforming it to numeric.



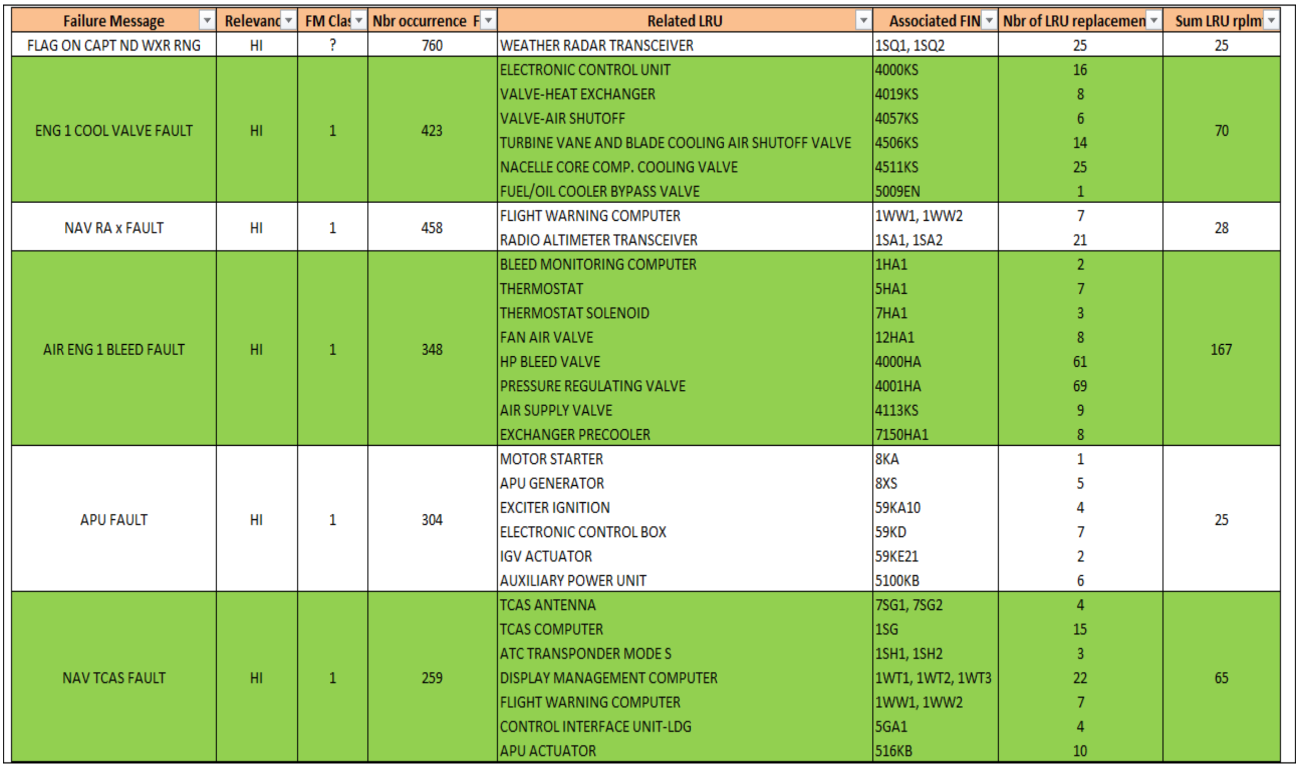
	EVENT_DATE	TAIL_NUMBER_ID	ATA	ATA_FIN_ID	SOURCE_ID	TEXT_ID	LEG_ID	FLIGHT_NUMBER_ID	4000HA	0	...	18	19	20	21	22	23	24
11551	1.488240e+18	0	3620.0	0	21	0	-3816	122	0	1	...	0	0	0	0	0	0	0
11552	1.488240e+18	0	3620.0	0	21	0	-3816	122	0	1	...	0	0	0	0	0	0	0
11553	1.489450e+18	0	3620.0	0	25	0	-3800	122	1	1	...	0	0	0	0	0	0	0
11554	1.487549e+18	1	3620.0	0	21	0	-3846	122	0	1	...	0	0	0	0	0	0	0
11555	1.490314e+18	1	3620.0	0	4	0	-3818	122	0	1	...	0	0	0	0	0	0	0

5 rows × 37 columns

Figure 2- 9 Snapshot of the ACMS data after transforming to numeric

2.4 Analysis of Intrinsic Characteristics in ACMS Dataset

First, the ACMS data is grouped based on failure messages, and associated components were determined. Figure 2-10 shows how the LRU replacement and its associated FIN's are grouped based on related failure messages X. For example, the failure message 'ENG 1 COOL VALVE FAULT' is highly associated with the following FIN's 4000KS,4019KS, 4057KS, and more.



Failure Message	Relevanc	FM Clas	Nbr occurrence F	Related LRU	Associated FIN	Nbr of LRU replacemen	Sum LRU rplm
FLAG ON CAPT ND WXR RNG	HI	?	760	WEATHER RADAR TRANSCEIVER	1SQ1, 1SQ2	25	25
ENG 1 COOL VALVE FAULT	HI	1	423	ELECTRONIC CONTROL UNIT	4000KS	16	70
				VALVE-HEAT EXCHANGER	4019KS	8	
				VALVE-AIR SHUTOFF	4057KS	6	
				TURBINE VANE AND BLADE COOLING AIR SHUTOFF VALVE	4506KS	14	
				NACELLE CORE COMP. COOLING VALVE	4511KS	25	
				FUEL/OIL COOLER BYPASS VALVE	5009EN	1	
NAV RA x FAULT	HI	1	458	FLIGHT WARNING COMPUTER	1WW1, 1WW2	7	28
				RADIO ALTIMETER TRANSCEIVER	1SA1, 1SA2	21	
AIR ENG 1 BLEED FAULT	HI	1	348	BLEED MONITORING COMPUTER	1HA1	2	167
				THERMOSTAT	5HA1	7	
				THERMOSTAT SOLENOID	7HA1	3	
				FAN AIR VALVE	12HA1	8	
				HP BLEED VALVE	4000HA	61	
				PRESSURE REGULATING VALVE	4001HA	69	
				AIR SUPPLY VALVE	4113KS	9	
				EXCHANGER PRECOOLER	7150HA1	8	
APU FAULT	HI	1	304	MOTOR STARTER	8KA	1	25
				APU GENERATOR	8XS	5	
				EXCITER IGNITION	59KA10	4	
				ELECTRONIC CONTROL BOX	59KD	7	
				IGV ACTUATOR	59KE21	2	
				AUXILIARY POWER UNIT	5100KB	6	
NAV TCAS FAULT	HI	1	259	TCAS ANTENNA	7SG1, 7SG2	4	65
				TCAS COMPUTER	1SG	15	
				ATC TRANSPONDER MODE S	1SH1, 1SH2	3	
				DISPLAY MANAGEMENT COMPUTER	1WT1, 1WT2, 1WT3	22	
				FLIGHT WARNING COMPUTER	1WW1, 1WW2	7	
				CONTROL INTERFACE UNIT-LDG	5GA1	4	
				APU ACTUATOR	516KB	10	

Figure 2- 10 Link between use cases and available data

The ACMS dataset reveals that a group of related failure messages (FM) can either result in LRU replacement (Positive class or labelled as 1) or Non-replacement (Negative class or labelled as zero). Then individual FINs are then plotted against their related failure messages.

2.4.1 Data Normality Check

Data normality check is useful in machine learning modelling because it helps in determining whether a data follows normal distribution or not. Knowing data normality can help determine the type of machine learning algorithm to use or develop for training the dataset for optimal modelling results. In order to test for data skewness, a Shapiro-walks test [5] is used. The test is calculated by multiplying the square of a suitable linear combination of sample order statistics by the standard symmetric estimate of variance. Because this ratio is invariant in both scale and origin, it can be used to test the composite hypothesis of normality. The hypothesis for the test is defined as when data is sampled from a Gaussian distribution; if the p-value is less than 0.05, it means the data is skewed. The result of the test is as follows P-Value = 0.044474. Since the P-value is less than 0.05, we reject the null hypothesis that we have sufficient evidence that the sample is not normally distributed.

Visual normality check is performed by creating histogram plots. Figure 2- 11 shows the histogram of the ACMS data, and the data shows mixed Gaussian distribution skewed distribution. Therefore, a log transform is applied to fix the skewness to make a good decision by machine learning model, as seen in Figure 2-12.

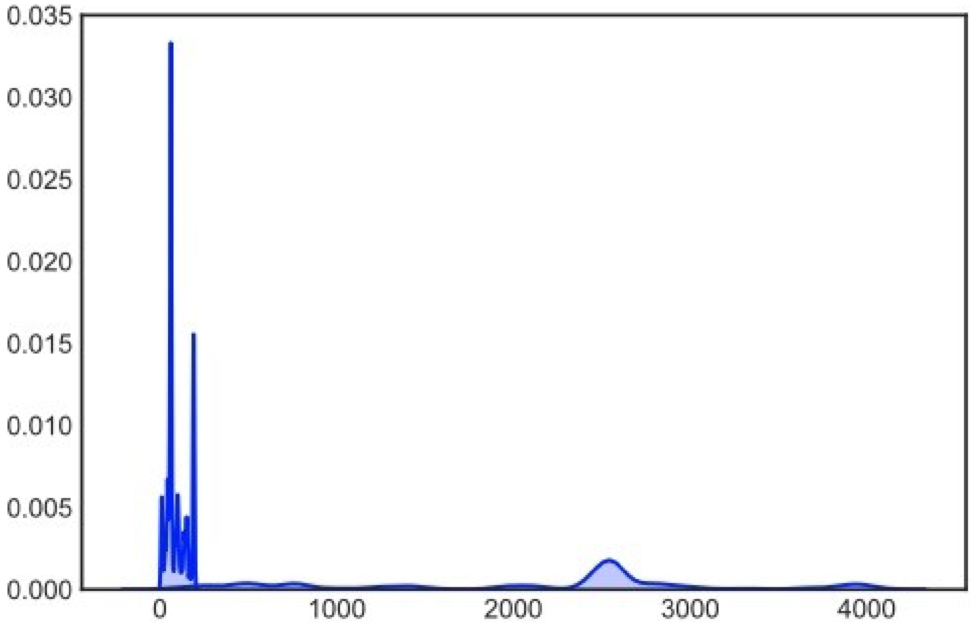


Figure 2- 11 Histogram before data transformation

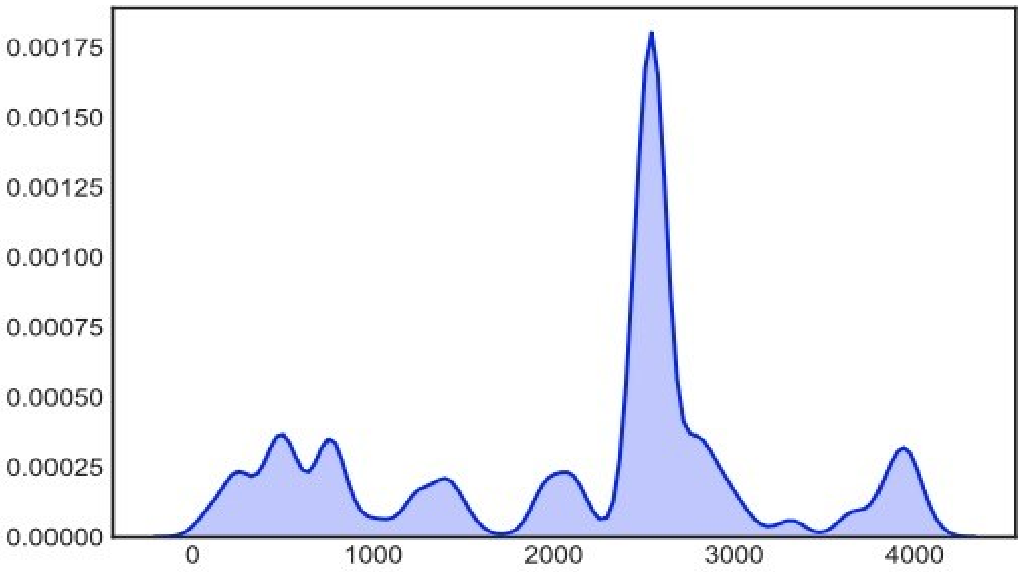


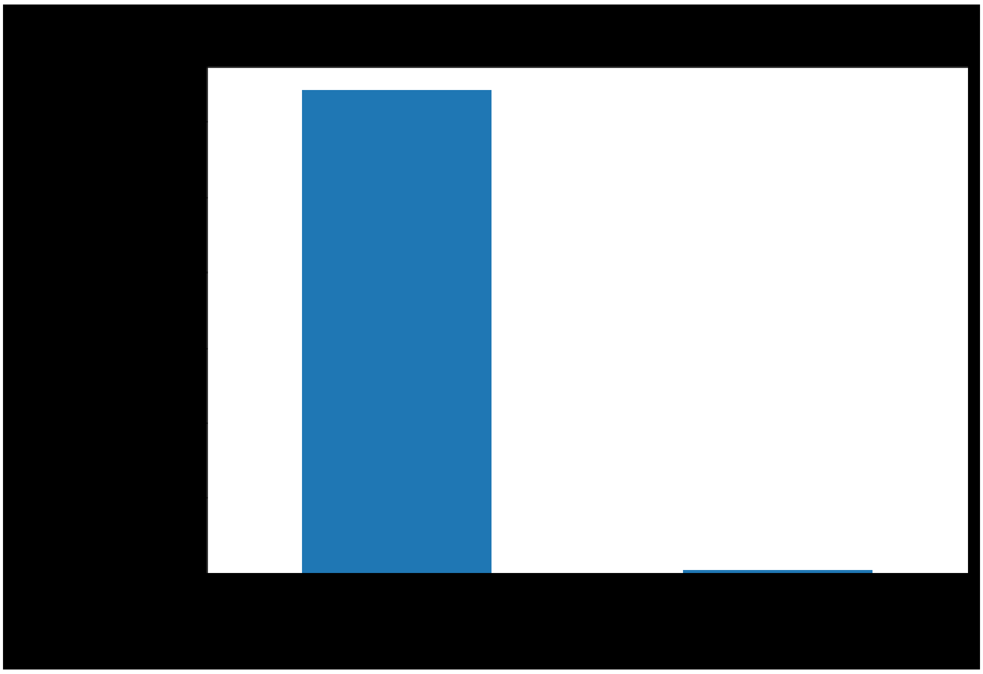
Figure 2- 12 Histogram after applying decomposition and log transformation

2.4.2 Scatter Plot Visualisation of ACMS Dataset

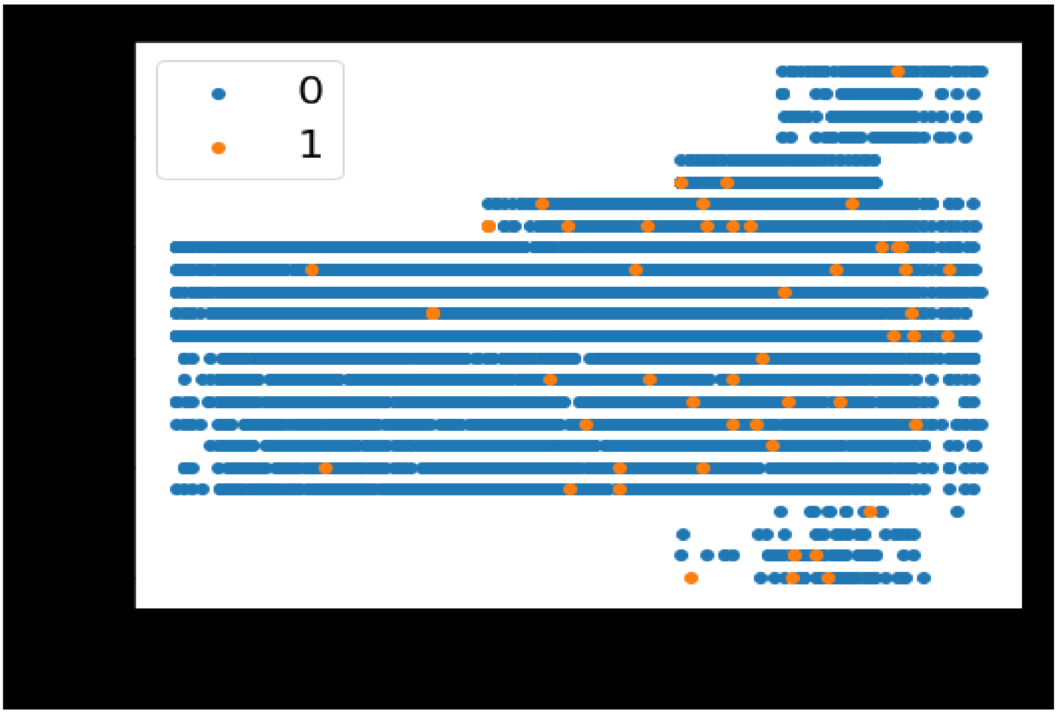
A dataset can be categorised as unbalanced if the distribution of classes is unequal. On the other hand, imbalanced data is commonly acknowledged in the machine learning community to refer to datasets with significant, and in some cases extreme, imbalances. A between-class imbalance is a form of imbalance in which one class vastly outnumbers another. Between-class imbalances of 100:1, 1000:1, and 10000:1 are not uncommon. To visually illustrate the imbalanced learning problem's real-world implications, consider the challenge of categorising component failure in an aviation system, where non-failure (labelled as negative or the majority class) and failure (labelled as positive or the minority class) exist.

Figures 2-13 to 2- 20 shows a scatter plot of all the selected components. Before generating the scatterplot for each component, the warning/failure signals associated with each component are grouped and converted to a two-dimensional numpy array. Figures 2-13 to 2- 20 - (a) depict the class distribution, while the (b) depicts the Scatter Plot. The components are represented by the Y-axis, while the X-axis represents the independent variable for the period under consideration. Non-failure (negatives) is represented by the blue dots, whereas the orange dots (positives) represent failure.

Subconcept, which appears in the distributions, which is of interest. By inspecting Figures 2-13 to 2-20, we see that both distributions exhibit relative imbalances, the between-class imbalanced where class (C0) outnumber class (C1). Also, the scatterplots in Figures 2-13 (b) have multiple concepts and severe overlapping. Due to a lack of representative data, some inducers may be unaware of specific concepts; this issue embodies imbalances caused by unusual occurrences, which we will investigate further. Imbalance due to rare instances is typical of domains with a small number of minority class examples, i.e. when the target concept is uncommon. Regardless of the between-class imbalance, the lack of representative data will make learning difficult in this case. In addition, the minority concept may contain a subconcept with few examples, resulting in varying degrees of classification difficulties. This is due to a different type of imbalance known as within-class imbalance [6][7][8], which deals with the distribution of representative evidence for sub-concepts inside a class.



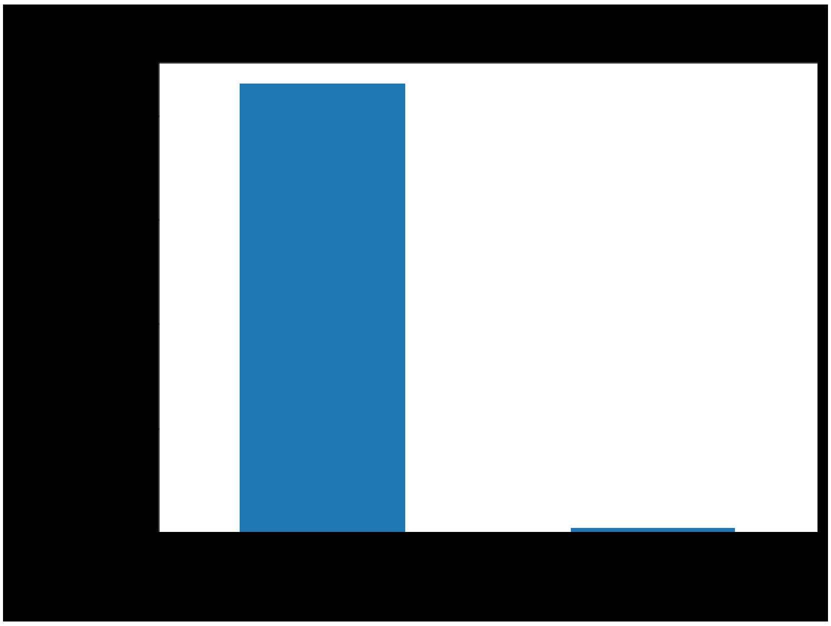
(a)



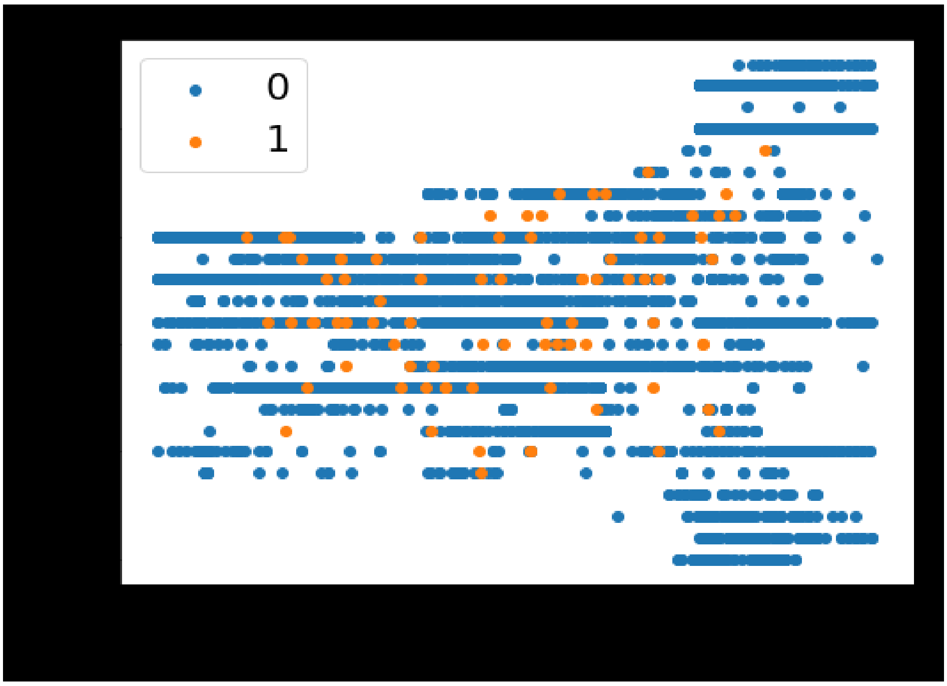
(b)

Figure 2- 13 (a) Class distribution and (b) the scatter Plot of 4000KS- electronic control unit replacement with points coloured by class value.

Data complexity includes overlapping, a lack of representative data, small disjuncts, and other issues. Figure 2- 13 illustrates this point. Some of the positive (orange) data points are mixed in with the negative (blue) data points (overlapping). Furthermore, some of the positive class instances are disjunct, which can cause traditional machine learning algorithms to generate an imperfect decision boundary. These challenges are again highlighted in Figures 2-14 to 2-20.

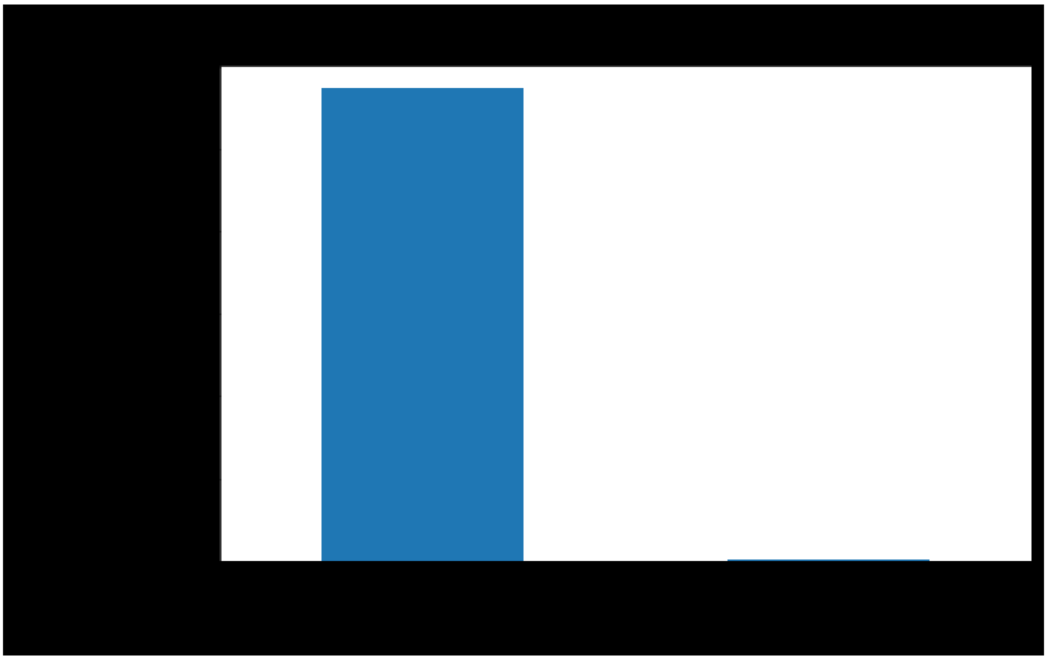


(a)

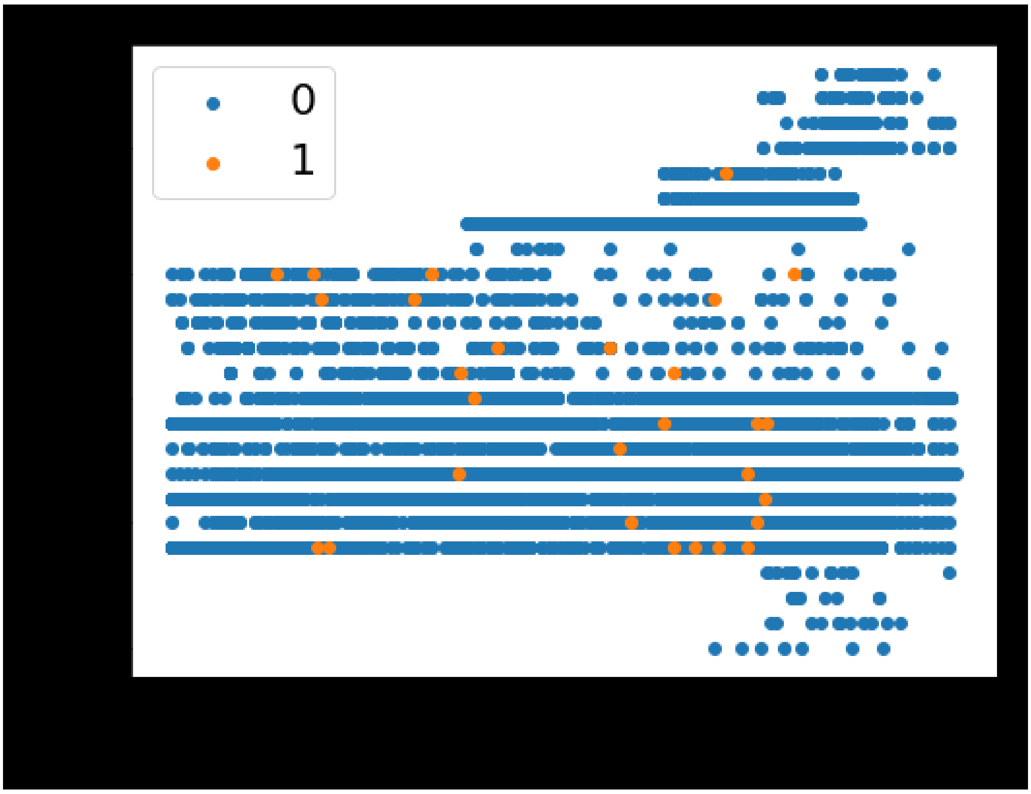


(b)

Figure 2- 14 (a) Class distribution and (b) the scatter Plot of 4000HA- high-pressure bleed valve replacement with points coloured by class value

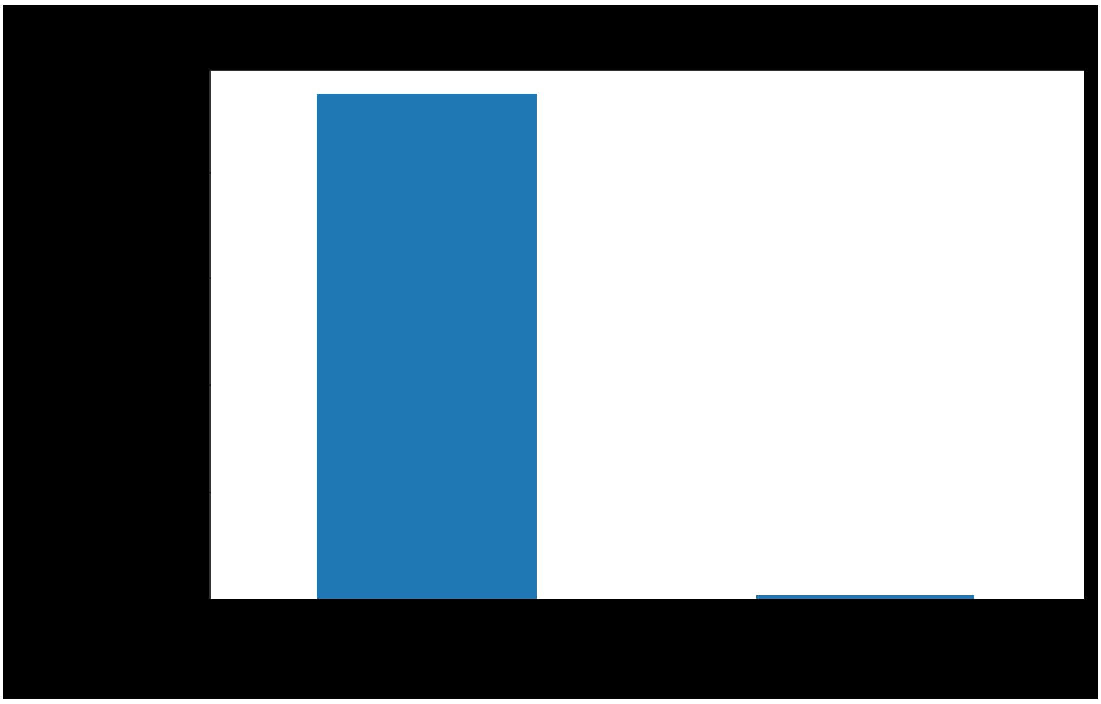


(a)

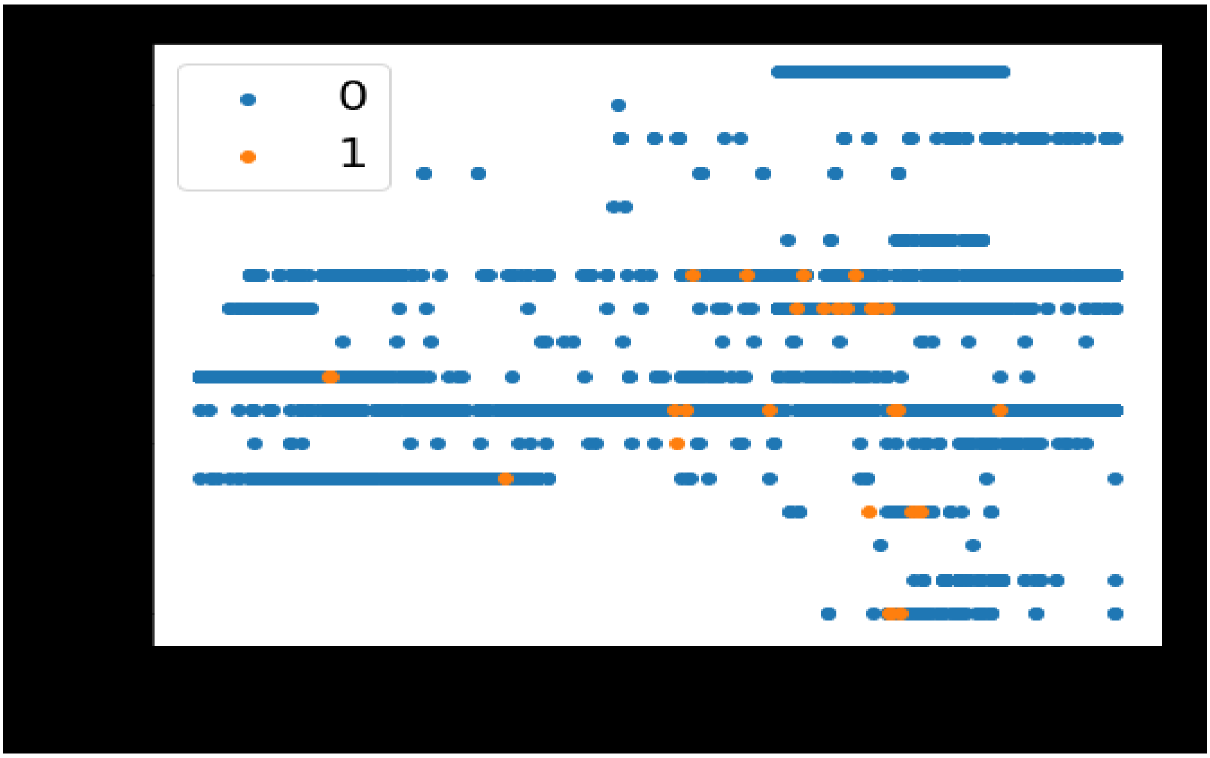


(b)

Figure 2- 15 (a) Class distribution and (b) the scatter Plot of 5RV1- satellite data unit replacement with points coloured by class value.

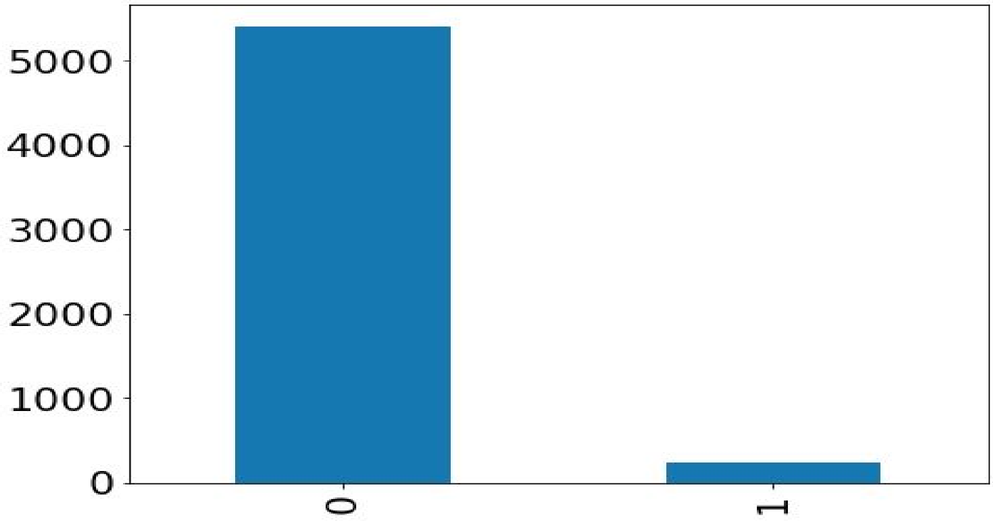


(a)

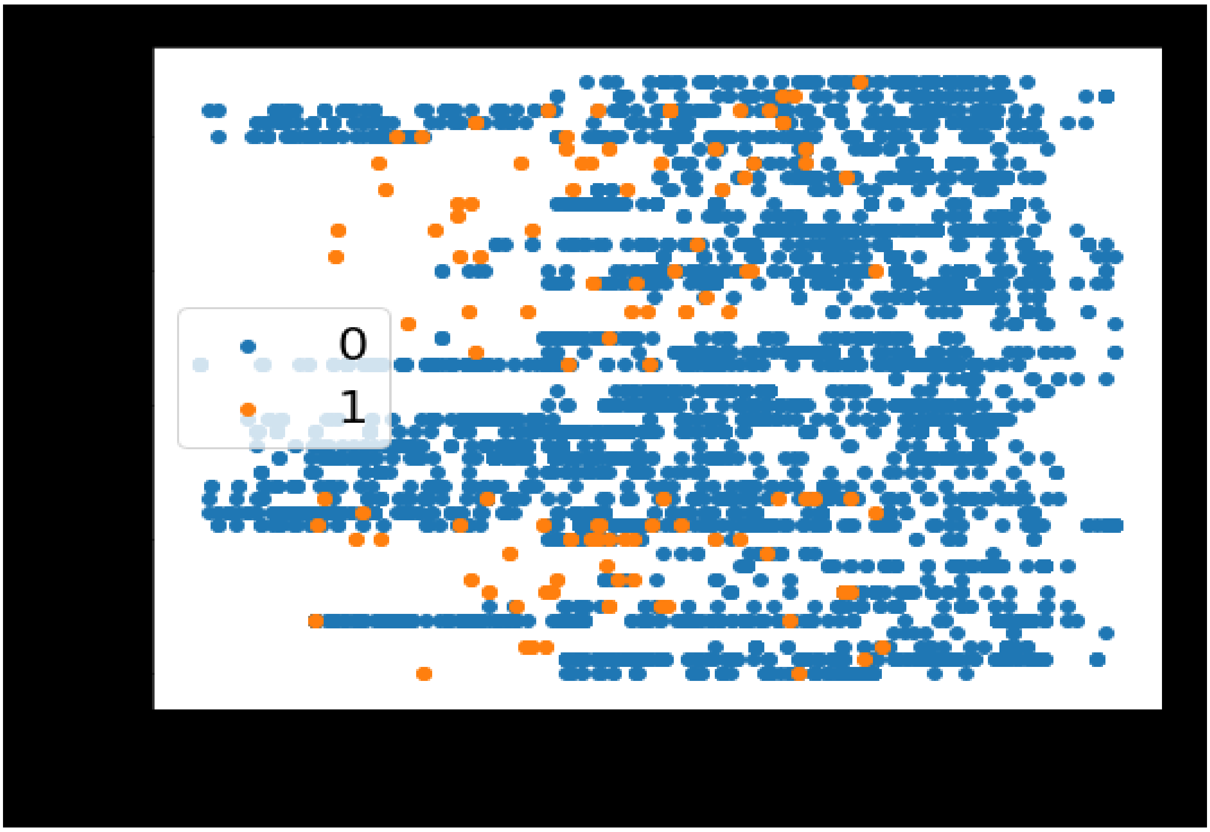


(b)

Figure 2- 16 (a) Class distribution and (b) the scatter Plot of 438HC- trim air valve replacement with points coloured by class value.

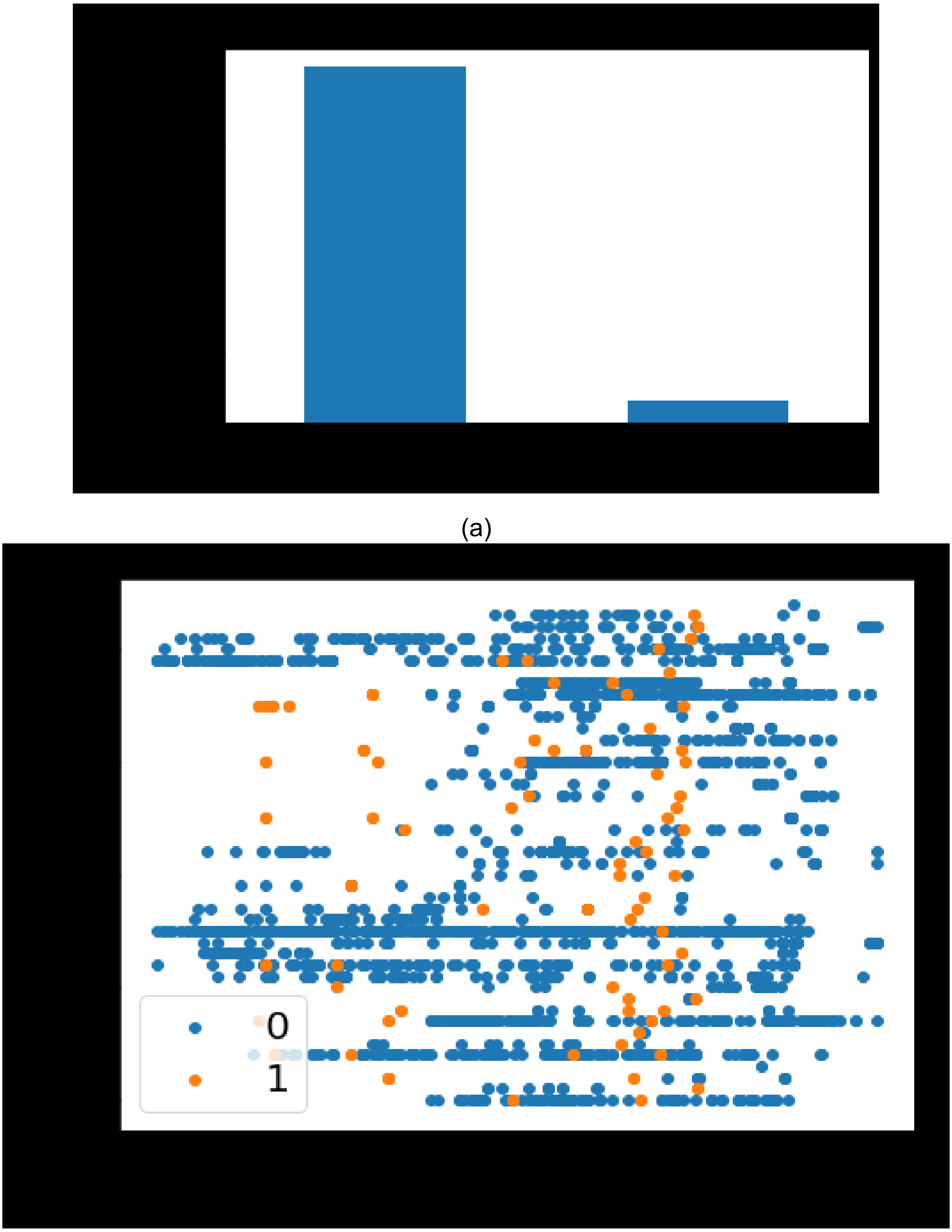


(a)



(b)

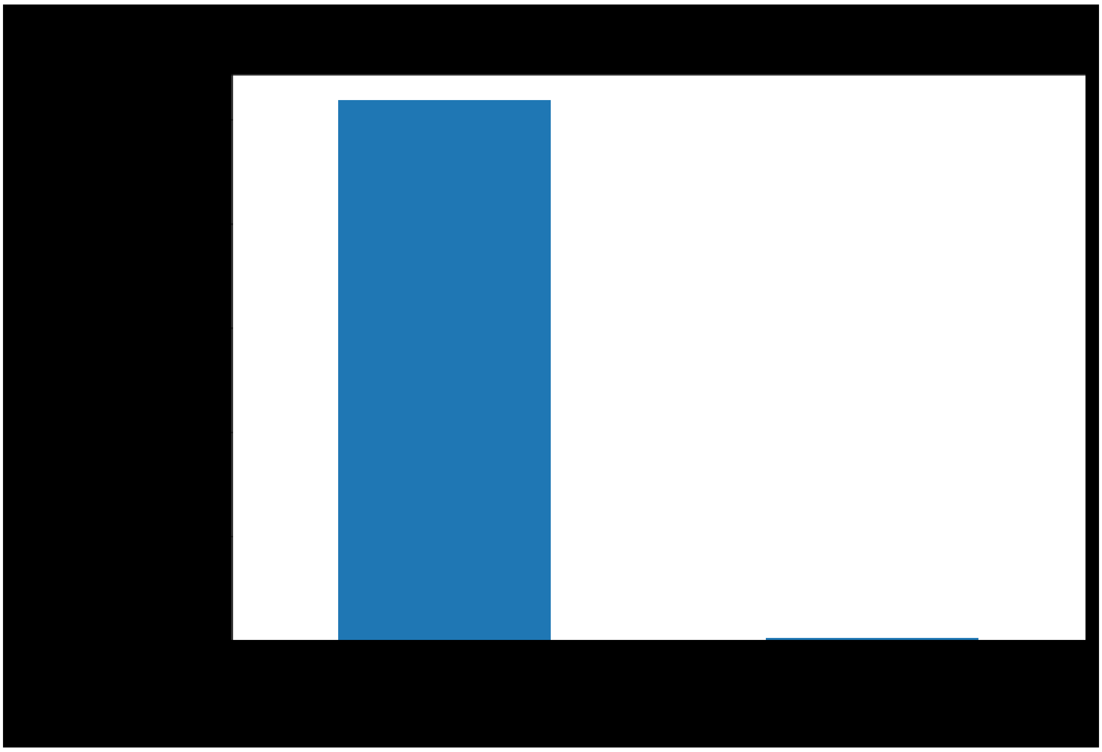
Figure 2- 17 (a)Class distribution and (b) the scatter Plot of 11HB- flow control valve replacement with points coloured by class value.



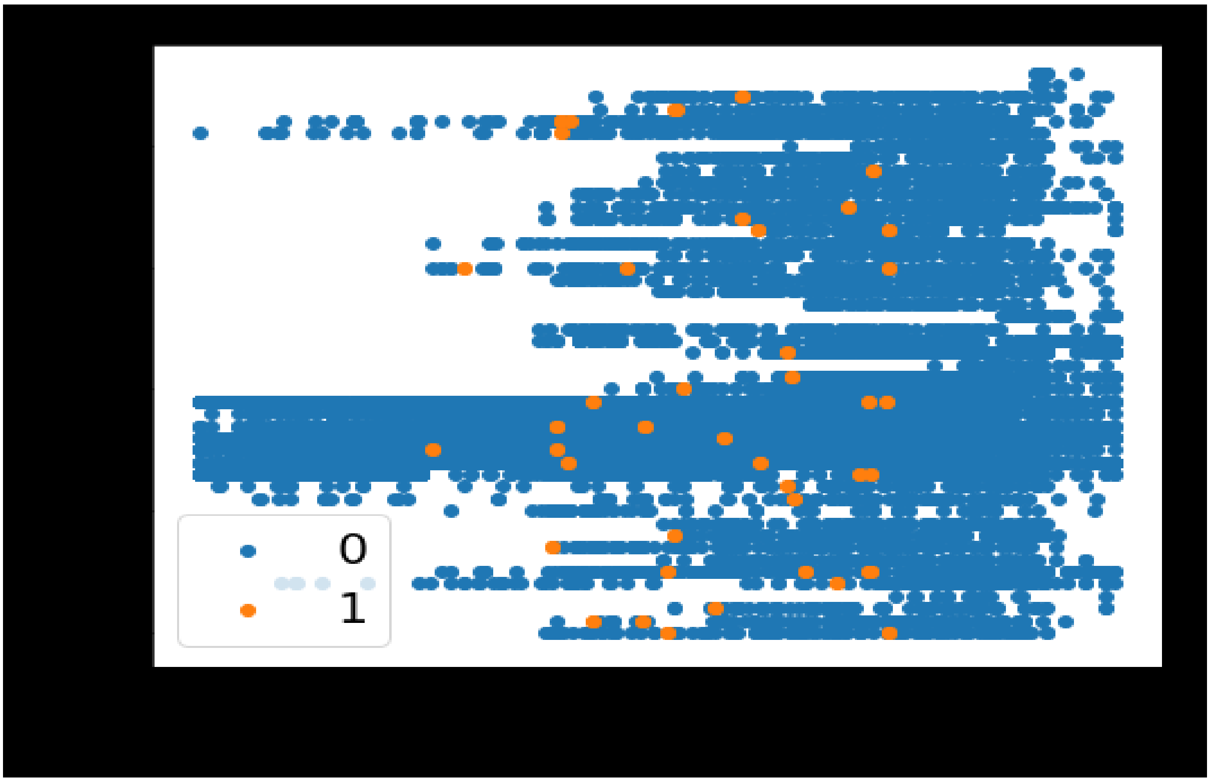
(a)

(b)

Figure 2- 18 (a) Class distribution and (b) the scatter Plot of 10HQ – avionics equipment ventilation computer replacement with points coloured by class value.

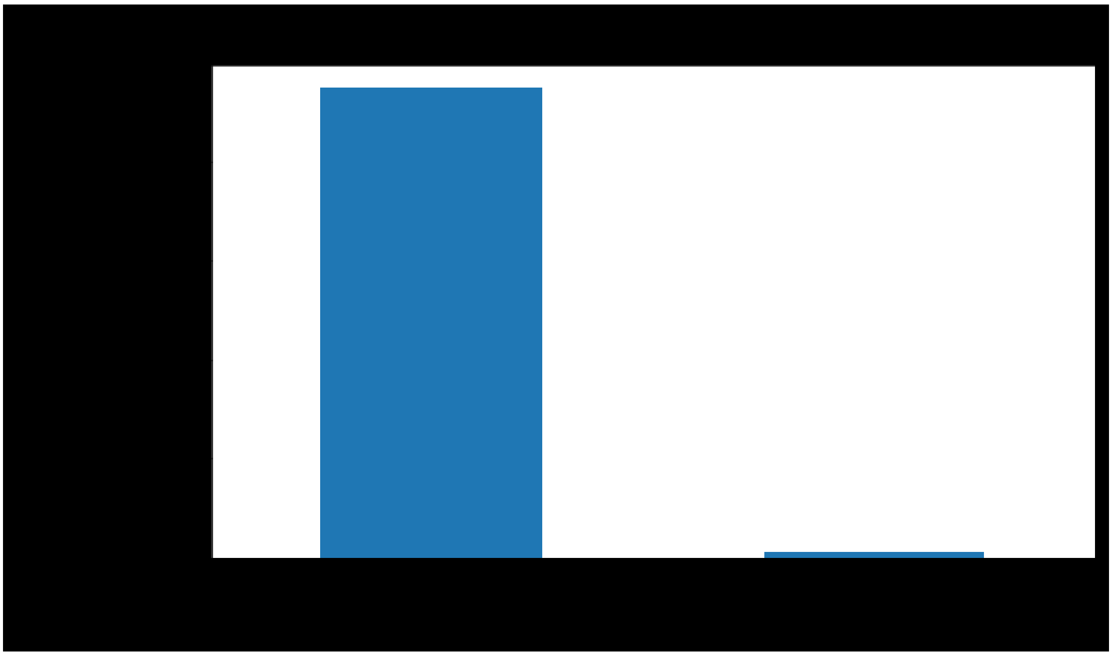


(a)

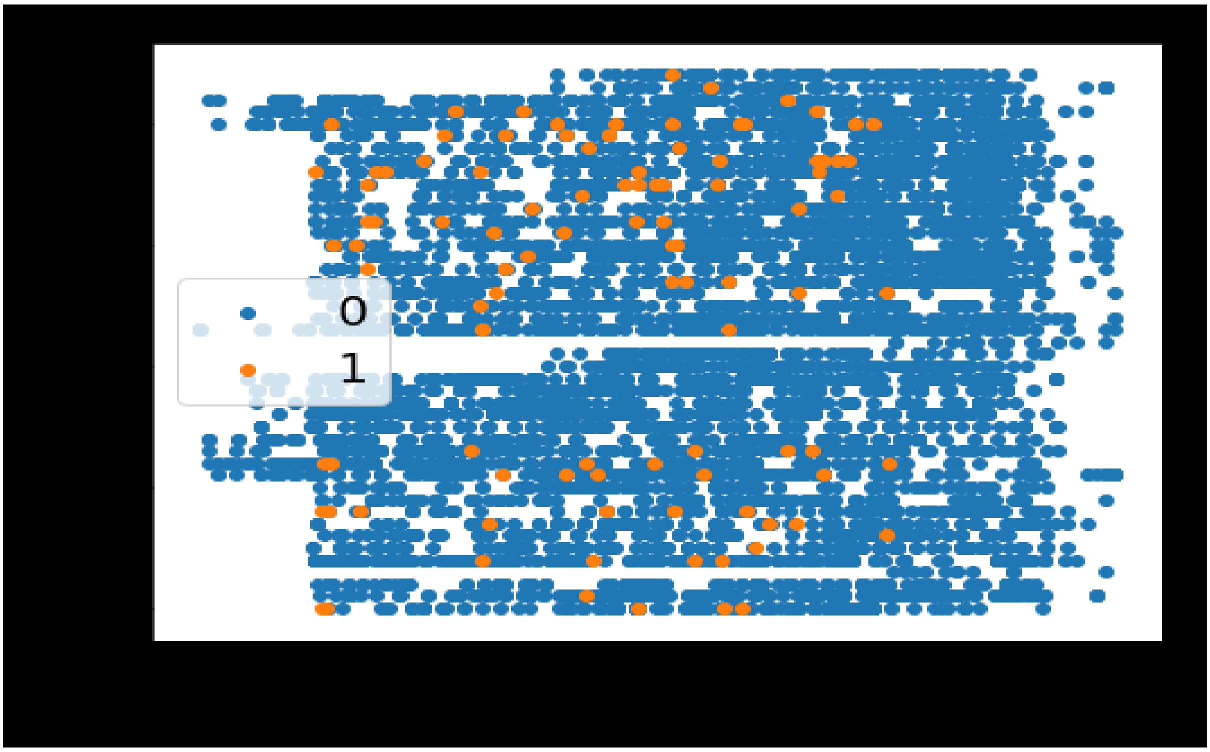


(b)

Figure 2- 19 (a) Class distribution and (b) the scatter Plot of 1TX1- air traffic service replacement with points coloured by class value.



(a)



(b)

Figure 2- 20 (a) Class distribution and (b) the scatter Plot of 8HB-Flow control valve-2 replacement with points coloured by class value.

Generally, the analysis of each component failure, looking at Figures 2-13 to 20, The proportion of positive label class and negatively label class (pattern not leading to failure) in all of the scatterplots indicates how the data is significantly skewed and overlapped. In machine learning and data mining research, data imbalance is a difficult problem to solve [9]. Many real-world data mining applications necessitate the creation of prediction models from datasets with very skewed distributions. Apart from the high imbalance that affects model performance, other data intrinsic characteristics such as small class disjunct and class overlapping can also cause performance loss. The overlapping problem occurs when some positive samples are mixed together with negative samples. As a result, a machine-learning algorithm may generate an inaccurate decision boundary, resulting in decreased model performance. In conclusion, the ACMS data is significantly skewed, as evidenced by the scatter plot depiction, with evidence of class overlaps, with so few positive examples and their unstructured nature, the significant class imbalance could provide a solid platform for utilising effective classification methods.

Another factor to consider is when the severely imbalanced data is combined with class overlap and the problem of a small sample size. Special methods are required to train such data for the predictive models. Large system logs face identification via image classification, and gene expressions are just a few examples of data with high dimensionality, small sample sizes, and abnormalities problems common in today's data analysis and knowledge discovery applications. The small sample size with extremely imbalanced problems has long been a research focus in the pattern recognition community [10][11]. The combination of imbalanced data, class overlapping and small sample size, on the other hand, poses a new difficulty to the community when the representative datasets' concepts display imbalances of the sorts outlined above. Two major difficulties arise concurrently in this case. Because the sample size is so small and intermixed, all of the difficulties around absolute rarity and within-class imbalances apply. Second, and more significantly, learning algorithms frequently fail to generalise inductive rules throughout the sample space when faced with this type of imbalance. Because of the difficulties in building conjunctions across the features with few samples, the combination of small sample size and high dimensionality in this scenario inhibits learning. If the sample space is large enough, a set of universal (though difficult) inductive rules for the data space can be defined. When samples are restricted, however, the rules that are formed can become overly specific, resulting in overfitting.

2.4.3 Density Plots

A density plot is a useful tool for visualising correlations between variables in data. A density map can be used to represent the proportion of data points that belong to a single variable. By superimposing different density graphs on top of each other, you can examine whether they overlap or not.

2.4.3.1 Visualization of a Probability Density Distribution

In probability theory, a normal distribution, also known as Gaussian or Laplace-Gauss, is a continuous probability distribution representing the real value of random variable variables whose distribution is unknown [13][14]. Two parameters define a normal distribution. Mean (μ) is the expected value of the distribution (and its median and mode), which controls the Gaussian distribution centre. Standard deviation (σ) corresponds to the expected square deviation from the mean, typically referred to as variance(σ2), which controls the shape of the distribution. The normal distribution is represented as N (μ, σ2), where a standard normal distribution with mean equals zero and variance equal to one can be described as N (0,1). Therefore, a probability density function (PDF) of a univariate normal distribution can be calculated when **mean** and **variance** are given. For a given value x, the dense is represented as follows.

$$P(x|\mu,\sigma^2) \; = \; \frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{(x-\mu)^2}{2\sigma^2}) \qquad (2\text{- }1)$$

Equation 2-1 is called a univariate normal distribution because it consists of one variable. Plotting the univariate PDF for the imbalanced ACMS data. Considering the data as a binary class (C0 and C1). Data points from class C1 follow a one-dimensional normal distribution of mean 0 and variance 5, and the data point from class C0 has a mean of C1. A plot in Figure 2-21 shows the data points along with the distribution of each class. As observed, the curve for class C0 is above the curve for class C1 in the high-density region (this is the probability, not the actual number of each class). The number of class C0 will be many times the number for class C1 because the dataset is imbalanced, meaning that for any given data point, the probability that the point is drawn from class C0 will always be greater than that drawn from class C1. The Plot clearly shows the effect of imbalance and how it can lead to a situation where the machine learning classifier can always classify examples from class C0, becoming biased to the negatively labelled (majority) class.

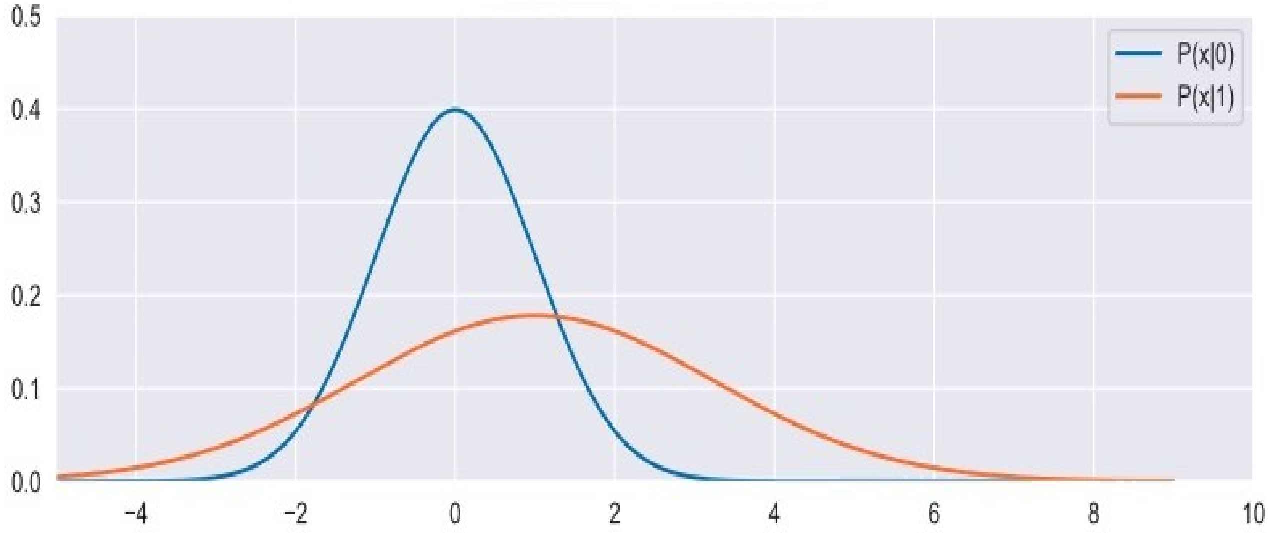


Figure 2- 21 Probability Density of Each class independently

2.4.3.2 Multivariate Gaussian Distribution

What is a multivariate Gaussian or normal distribution, and how does it differ from a single-variable normal distribution? The term "multivariate" refers to the presence of many variables. Our goal is to portray a normal distribution in several dimensions. Multivariate Gaussian Distribution represents the distribution of a multivariate random variable (a multi-dimensional generalization of the one-dimensional Gaussian distribution); it comprises a correlated random variable. The Central Limit Theorem [15], states that a multivariate distribution develops from sums of random variables under general conditions, something often borne out, at least approximately, by actual data, which partially justify the focus on multivariate Gaussian distributions [16]. The multivariate Gaussian distribution is represented by the following parameters [13].

Mean (μ): which is the expected value of the distribution. Represented as d X 1 mean vector.

Variance or covariance matrix Σ: which shows how random variables depend on each other and how the variables change together. The matrix is of size $d\ X\ d$.

The multivariate Gaussian distribution can be denoted as M (μ, Σ). The covariance between random variables $\boldsymbol{X1}$ and $\boldsymbol{X2}$ can be represented as $COV\ (x1, x2)$ [17].

A Join probability density of a multivariate Gaussian with dimension d is given as

$$P(x|\mu,\Sigma\,) \;=\; \frac{1}{2(\pi)^d|\Sigma|}\exp(-\frac{1}{2}\,(x-\mu)^t\Sigma^{-1}\,(x-\mu) \qquad (2\text{- }2)$$

Where x is a random variable of size d and $|\,\Sigma\,|$ is the determinant. The covariance matrix Σ, which represents the variances of all possible pairs of variables as well as the covariances between them. Given an n-dimensional random vector, for simplicity, say two dimensions:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad (2\text{- }3)$$

which has a normal distribution M (μ, Σ) were

$$\mu = \begin{bmatrix} \mu_{\mathbf{1}} \\ \mu_{\mathbf{2}} \end{bmatrix} \qquad (2\text{- }4)$$

and covariance

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \qquad (2\text{- }5)$$

The multivariable can be independent or dependent correlated, which is represented as

$$\text{M}\ (\mu,\ \Sigma) = (\ \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}) \qquad (2\text{-}6)$$

If the variables are independent, then the covariance will be 0. For example, the mean and variance for random variable X_1 and X_2 is represented as

$$M(\begin{bmatrix}0\\1\end{bmatrix},\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix})$$

If the variables are correlated, the covariance is set to the respective values other than zero. The following parameters are obtained by calculating the mean and variance of the ACMS data.

$$M(\begin{bmatrix}0\\1\end{bmatrix},\begin{bmatrix}1 & 0.7\\0.3 & 1\end{bmatrix})$$

An N-dimensional array is created from the ACMS dataset. Then, a multivariate Gaussian is plotted for independent and correlated variables, as illustrated in Figure 2-22. The multivariate Gaussian is useful because of its algebraic features, which ensures that we get a normal distribution when marginalising. Marginalisation is a method for determining the marginal contribution of another variable by summing the various values of one variable. The end result is the distribution of a subset of the variables without reference to the removed ones.

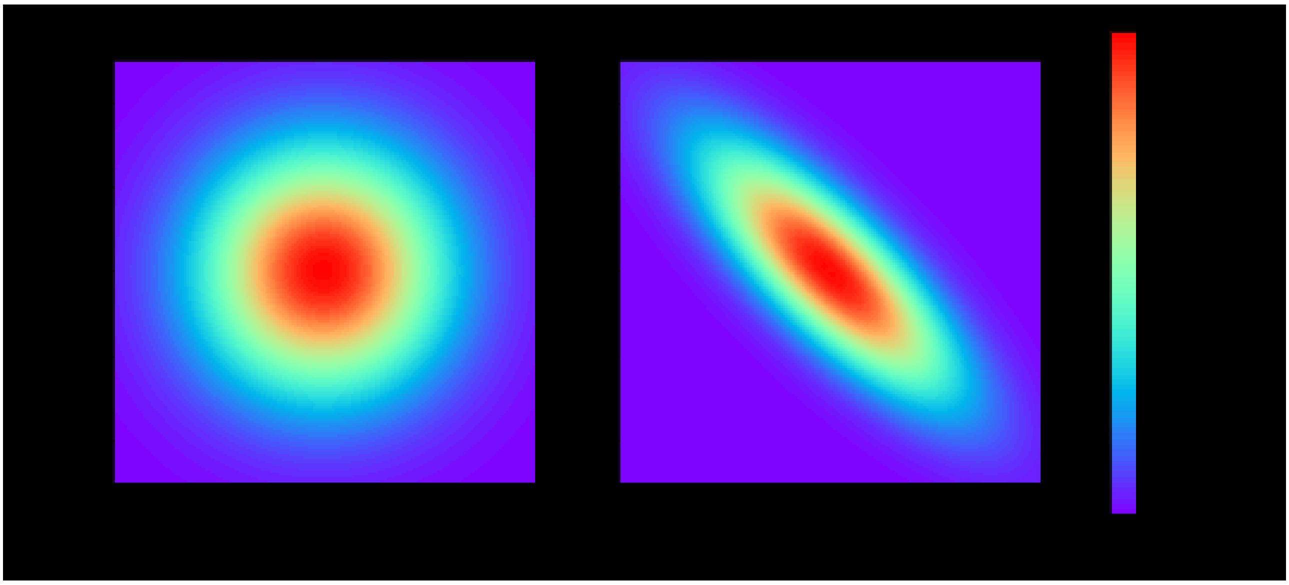


Figure 2- 22 multivariate Gaussian with dimension d X d for random variables X and Y. The figure on the left shows a multivariate Gaussian density for independent and the figure on the right for correlated variables

Using a multivariate Gaussian for correlated variables. The probability of component failure is plotted as red dots on the probability density surface, as shown in Figure 2-23.

The data is first of all transformed/ decomposed into a 2D array using a Numpy python library. The following parameters are obtained from the resulting array by calculating the mean and variance of the failure message and target component.

$X_{per} = M(\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0.7 \\ 0.3 & 1 \end{bmatrix})$

$Y_{per} = M(\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ -1 \end{bmatrix})$

Creating lower-triangular covariance L using Cholesky decomposition. then Apply the transformation Y = L.dot(X) + Y_{per}.

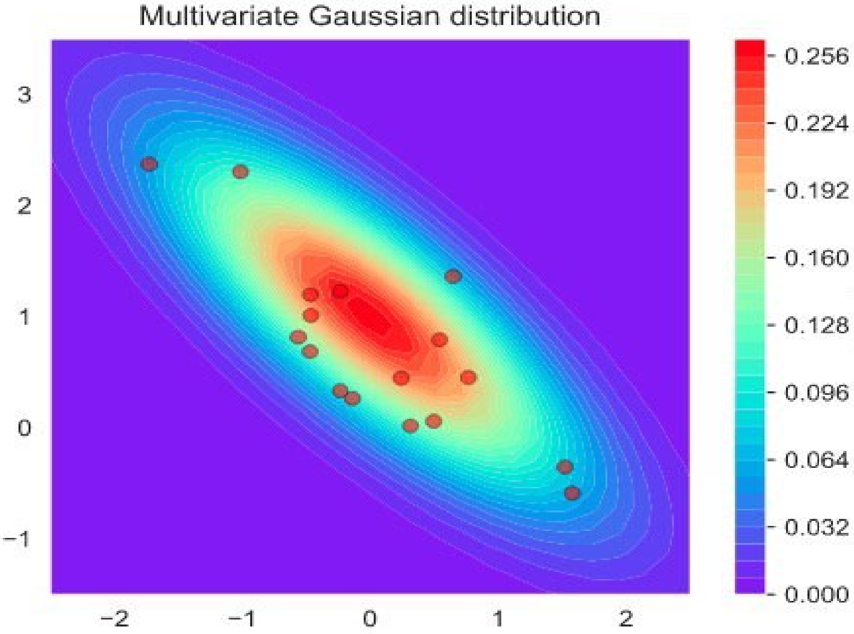


Figure 2- 23 Drawing the probability of component failure on the probability density surface.

It can be observed from Figure 2- 23 that the failure (component replacement) occurs majorly in the region with high density. This shows how challenging for a machine learning classifier to separate between classes effectively because of the ambiguity region separating the classes. It can so be observed that the rare cases formed small class disjunct.

One of the challenges with the conventional distance-based approach for this type of mixed data (numerical and categorical) is that the distribution will lose its true representation because of the encoded categorical variables. When categorical variables are present, the distance metric becomes meaningless, and they must be addressed differently than traditional distance-based methodologies.

Density plot of numerical variables

Numerical variables were selected, then the data was transformed into the 2D array using the NumPy array function. The mean and variance of both failures and the normal class are then calculated, and the probability of drawing failure is presented in Figure 2-24.

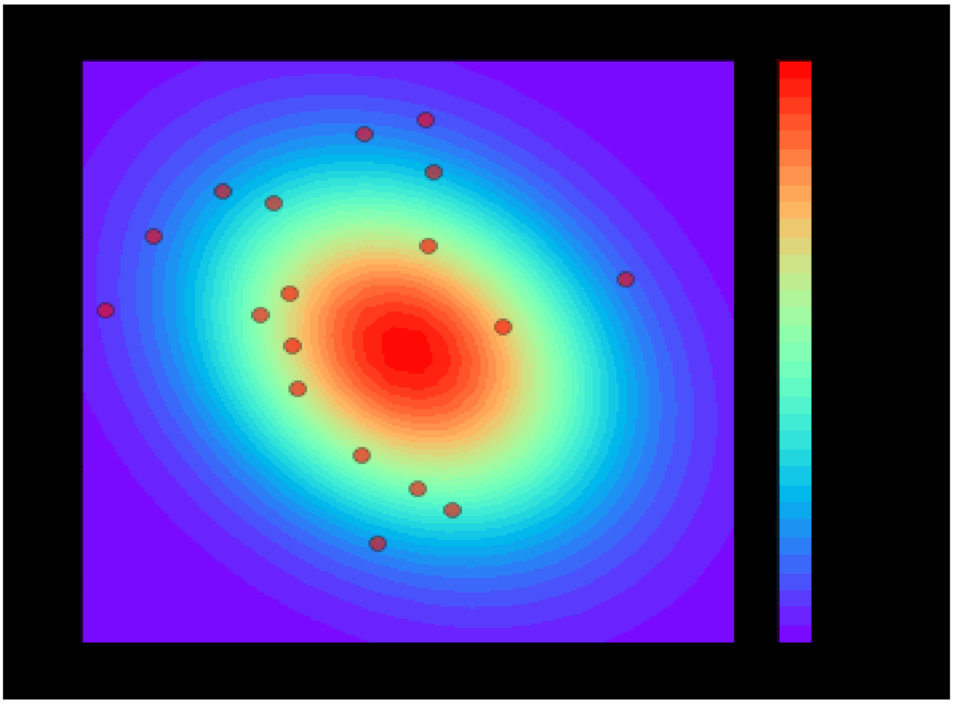


Figure 2- 24 Density plot for numerical values

Generally, the information obtained from the data visualization indicated that in the case of ACMS data, the data is extremely imbalanced, and classes are not well separated, which is challenging to use traditional machine learning. The data has a skewed class distribution, resulting in an extremely imbalanced, class overlapping, and class-disjunct problem. Apart from the extremely imbalanced problem in the ACMS dataset, it is also accompanied by the following intrinsic challenges: Class Separability-

Figure 2-23 shows how the classes are not well separable (overlapping). It can observe that facing imbalanced does not always mean the classes are not separated. Even when the classes are separated, if the probability of sampling from one class is more than the other, the classifier can always be biased to the class with high probability because the points are more likely to be drawn from the class with high probability. Hence, robust methods are required to handle such challenges.

2.5 Imbalanced Datasets Challenge to Traditional Machine Learning Algorithms

Imbalanced classification problems related to other dynamics and characteristics inherent in the dataset have been studied for several decades [20], yet several issues remain unresolved.

The reason why imbalanced classification is difficult for traditional machine learning algorithms, which leads to performing poorly, is that they are designed with some assumptions such as:

1. The classes in the dataset are balanced.

2. There is a significant number of data in each class

3. The class probability distributions of all the classes are the same

4. The data label is noise-free

5. The cost of misclassification is the same for both classes.

6. The sub-concepts present in each class are represented equally

Violating one or more of the assumptions mentioned above is referred to as a data irregularities problem, leading to performance degradation in data-driven machine learning models [21]. There exist many domains that the data violates one or more of the above assumptions. An example of such a domain is the ACMS data (as visualized), which is extremely imbalanced, with unequal sub-concepts within the classes. Also, considering the use of the dataset for predictive aircraft maintenance modelling, the cost of misclassification is not the same for both classes. For example, false positives are more critical (can lead to high loss or fatality) than false negatives (only increases maintenance checks). Thus, this study focuses on the imbalanced learning approach for rare failure prediction using the ACMS data. Particularly on the impact of severe class Imbalance, small disjuncts, class-overlapping on classifier performance.

2.5.1. Class Overlapping and Class Separability

Class overlapping is caused due to ambiguous regions in the data where the prior probability of two or more classes is approximately equal, making it very difficult to distinguish between the two classes within the overlapping area. If the data is linearly separable and does not contain any intrinsic data challenges, some classifiers can easily classify it. In a situation where class overlapping exists and other inherent challenges, there is a need to address those intrinsic challenges to obtain a good machine learning model. Many works have focused on the study of the relationships between class imbalance and overlapping problems in predictive modelling. Particularly Vuttipittayamongkol P et al. [22] provided an intensive discussion about the impact of class overlapping on classifier performance. [22] the authors provided an intensive discussion about the impact of class overlapping on classifiers performance. They compared different oversampling techniques, mostly SMOTE and its derivatives, and concluded that classification errors increased with the degree of class overlap regardless of imbalance. Also, they state that the effect of class imbalance highly depended on the presence of class overlap. Also, In a study by Prati RC et al.[23].[23],the authors generated an artificial dataset and set up an experiment to show that the degree of class overlapping strongly correlates with class imbalance, varying the degree of overlapping and imbalance ratio between classes. They concluded that class probabilities are not the main responsibility for classifier performance degradation but rather the degree of overlap between classes. For further information on challenges and opportunities in imbalance learning, the reader can refer to Johnson JM et al. [3].

2.5.2. Class Small Disjuncts

Rare cases correspond to the minority class of the training set in a particular area of the feature space. In concept learning, rare cases are vital to consider because they cause the occurrence of a small class disjunct and are identified to be more error-prone than the large class disjunct [24]. More elaborately, machine learning algorithms usually create concepts made up of many disjunct. Each disjunct is, in turn, a conjunctive definition of the sub-concept of the original concept. The coverage of the disjunct corresponds to the number of the training examples that are correctly classified, and disjunct is considered small if the coverage is small otherwise large [24]. Small class disjuncts are not inherently error-prone than large disjuncts. What makes it error-prone is the classifiers biases [5] and other factors such as the noise, missing values, data size, and other factors [24].

Furthermore, a small disjunct arises when data in the same class is represented with different clusters of concepts (within class imbalance). Although those small disjuncts are implicit in most of the problems, and also cover few instances in a trained model and generally have much higher error rates in contrast to large disjuncts. The less represented small sub-clusters can further worsen classification performance degradation in an extreme imbalance dataset. It becomes hard to know whether these sub-concepts represent actual sub-examples or are merely attributed to noise [25]. Many works have focused on studying the relationship between small class disjunct and imbalanced classification problems. Notably, a broad discussion about the impact of class disjuncts as it relates to the class imbalance classification problem can be found in Das s et al.[21]. The authors reviewed the class disjunct problem and concluded that small class disjunct is the major cause of misclassification. Hence new research in the field must strive to outperform the current solutions. Other studies focus on the impact of combining class imbalance and small disjunct [9]. The authors show how extreme imbalance can give rise to the small class disjunct. Another has study focus on the impact of class imbalance and class skew distribution [21]. The open literature lacks a study on the impact of the combination of class Imbalance, small disjuncts and class distribution skew on classifier performance, especially for the log-based dataset.

2.6 Reference

1. Ferreira JF. AIRMES Newsletter 2020. 2020; (681858): 1–11. Available at: DOI:http://www.airmes-project.eu/files/newsletters/AIRMES_Newsletter4_September2019.pdf

2. Airbus. ACMS Discription Manual. Airbus; 2000. Available at: DOI:https://wenku.baidu.com/view/179923f4910ef12d2af9e723.html

3. Mostafa SM. Missing data imputation by the aid of features similarities. International Journal of Big Data Management. 2020; 1(1): 81. Available at: DOI:10.1504/ijbdm.2020.106883

4. Kuhn M., Johnson K. Applied Predictive Modeling with Applications in R. Springer. 2013. 615 p. Available at: http://appliedpredictivemodeling.com/s/Applied_Predictive_Modeling_in_R.pdf

5. Shapiro SS., Wilk MB. An Analysis of Variance Test for Normality (Complete Samples). Biometrika. 1965; 52(3/4): 591. Available at: DOI:10.2307/2333709

6. Ke H., Lu C., Xu H. Global cost parameter selection of extreme learning machine for imbalance learning. Harbin Gongcheng Daxue Xuebao/Journal of Harbin Engineering University. 2017; 38(9): 1444–1449. Available at: DOI:10.11990/jheu.201610045

7. Miao Z., Zhao L., Yuan W., Liu R. Multi-class imbalanced learning implemented in network intrusion detection. 2011 International Conference on Computer Science and Service System, CSSS 2011 - Proceedings. 2011. pp. 1395–1398. Available at: DOI:10.1109/CSSS.2011.5975051

8. Zhang X., Li Y., Kotagiri R., Wu L., Tari Z., Cheriet M. KRNN: k Rare-class Nearest Neighbour classification. Pattern Recognition. 2017; 62: 33–44. Available at: DOI:10.1016/j.patcog.2016.08.023

9. Moniz N., Monteiro H. No Free Lunch in imbalanced learning. Knowledge-Based Systems. Elsevier B.V.; 2021; 227: 107222. Available at: DOI:10.1016/j.knosys.2021.107222

10. Stefanowski J. Challenges in Computational Statistics and Data Mining. 2016. Available at: DOI:10.1007/978-3-319-18781-5

11. Hu Y., Guo D., Fan Z., Dong C., Huang Q., Xie S., et al. An Improved Algorithm for Imbalanced Data and Small Sample Size Classification. J. Data Anal. Inf.

Process. 2015; 03(03): 27–33. Available at: DOI:10.4236/jdaip.2015.33004

12. He H., Garcia EA. Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. 2009; 21(9): 1263–1284. Available at: DOI:10.1109/TKDE.2008.239

13. Roussas. Probability Density Functions in One Variable Distribution Binomial,. 2003.

14. Sherlock A., Hume B. An Introduction to Probability and Statistics. The Mathematical Gazette. 1968. 68 p. Available at: DOI:10.2307/3614484

15. Kwak SG., Kim JH. Cornerstone of Modern Statistics. Korean Journal of Anesthesiology. 2017; 70(2): 144–156.

16. So I., Pdf M. Chapter 4 Multivariate Random Variables, Correlation, and Error Propagation. 2008; (1994): 1–14. Available at: DOI:https://igppweb.ucsd.edu/~agnew/Courses/Sio223a/sio223a.chap4.pdf

17. Vosshenrich R., Doler W., Hellige G., Muller E., Hausmann R., Fischer U., et al. Einsatz Der Race-Technik Zur Quantitativen Flussmessung. Evaluierung an Einem Klinisch Relevanten Flussmodell. RoFo Fortschritte auf dem Gebiete der Rontgenstrahlen und der Neuen Bildgebenden Verfahren. 1993; 158(6): 550–554.

18. Das B., Krishnan NC., Cook DJ. Handling class overlap and imbalance to detect prompt situations in smart homes. Proceedings - IEEE 13th International Conference on Data Mining Workshops, ICDMW 2013. 2013; : 266–273. Available at: DOI:10.1109/ICDMW.2013.18

19. López V., Fernández A., García S., Palade V., Herrera F. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences. Elsevier Inc.; 2013; 250: 113–141. Available at: DOI:10.1016/j.ins.2013.07.007

20. Johnson JM., Khoshgoftaar TM. Survey on deep learning with class imbalance. Journal of Big Data. Springer International Publishing; 2019; 6(1). Available at: DOI:10.1186/s40537-019-0192-5

21. Das S., Datta S., Chaudhuri BB. Handling data irregularities in classification: Foundations, trends, and future challenges. Pattern Recognition. Elsevier Ltd; 2018; 81: 674–693. Available at: DOI:10.1016/j.patcog.2018.03.008

22. Vuttipittayamongkol P., Elyan E., Petrovski A. On the class overlap problem in imbalanced data classification. Knowledge-Based Systems. Elsevier B.V.; 2021; 212: 106631. Available at: DOI:10.1016/j.knosys.2020.106631

23. Prati RC., Batista GEAPA., Monard., C. M. Class imbalances versus class overlapping: an analysis of a learning system behavior. MICAI 2004 Adv. Artif. Intell. 2004; (0): 312. Available at: https://link.springer.com/chapter/10.1007/978-3-540-24694-7_32

24. Jo T. Class Imbalances versus Small Disjuncts. 6(1): 40–49. Available at: DOI:https://dl.acm.org/doi/pdf/10.1145/1007730.1007737

25. Bauder RA., Khoshgoftaar TM., Hasanin T. An Empirical Study on Class Rarity in Big Data. Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018. IEEE; 2019; : 785–790. Available at: DOI:10.1109/ICMLA.2018.00125

CHAPTER 3: Review of Imbalanced Learning Methods and The Application of Predictive Maintenance

This chapter provides an overview of data analytics in the aerospace sector, with a focus on imbalance learning for maintenance forecasting. It also goes through the issues and possibilities of leveraging data from the aircraft central maintenance system to construct predictive maintenance models. The review is separated into three parts, as indicated in Figure 3-1. The first section explores techniques for dealing with data imbalances. The second section looks at how data analytics can be used in the aerospace industry in general. The final section delves into predictive maintenance modelling and offers an overview of maintenance approaches.

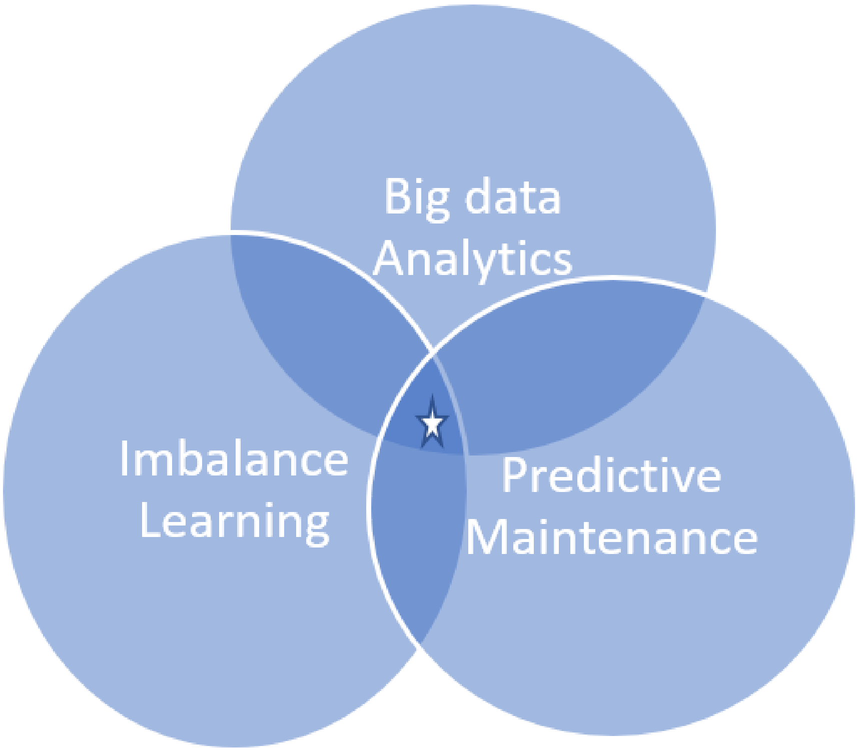


Figure 3- 1 Map of Literature Review

3.1 Imbalance learning

Data imbalance is a difficult problem in machine learning and data mining research. When the distribution between classes in a dataset is unequal, it is considered imbalanced. Many real-world data mining applications necessitate the creation of prediction models from datasets with very skewed distributions [1]. With the introduction of Industry 4.0 and further advancements in data analytics, the creation, storage, and analysis of vast amounts of data have become more inexpensive. Vehicle maintenance procedures have also undergone substantial changes as a result of technological advancements. Shifting from preventive to predictive to prescriptive maintenance, for example. The study of data-driven prognostic modelling for aircraft maintenance is becoming increasingly popular [2,3]. However, one of the major issues facing data analytics researchers is the underrepresentation of failure behaviour in relation to target events. Due to the infrequent occurrence of failure events, resulting in a machine learning challenge known as an imbalanced classification problem [4]. This problem arises when the distribution of classes present in the dataset is not uniform. The total number of instances in one class far outnumber the other class (also known as a skewed distribution). Training machine learning algorithms with the imbalanced dataset can cause biases in classification, resulting in model performance degradation. Hence, producing a high rate of false-positive and also imprecise prognostics of vehicles failures. The imbalanced classification problem or rare event prediction problem is generally prevalent in many application domains. For example, the historical data is often imbalanced in aircraft operation because of the safety measures in place; the vehicle is expected to function normally with fewer faults. The rare failure also causes irregular patterns and trends in the generated dataset [5].

Data imbalanced problems or rare event prediction can also be seen in different domains. For example, in the financial sector, fraud detection, where illegitimate transactions are infrequent and irregular compared to large legitimate ones. Predicting the rare fraudulent ones is critical because it can cause enormous damage to business [6]. Similarly, in clinical science, rare event prediction is evident in the diagnosis and prognosis of rare diseases; in most situations, the healthy population significantly outnumbers the sick population [7]. Likewise, when detecting an oil leak in the ocean, satellite photos may display a few images reflecting the oil spillage section, while the majority of the images depict non-spill areas [8]. Rare event prediction can be seen in developing models to predict an earthquake's occurrence; since earthquakes rarely occur and with an irregular pattern. The literature shows that most of the solution to handling imbalance classification problems or rare failure predictions depends on the application domain or dataset. Therefore, handling an imbalanced classification problem using ACMS data in aircraft predictive maintenance modelling remains an open research issue [9–11].

3.1.1 Review of methods for handling extremely imbalanced Class overlapping and small class disjunct problems in the ACMS dataset.

In concept learning, the dataset is said to be imbalanced if examples of a class far outnumber others. Such a situation poses a challenge for traditional machine learning algorithms such as support vector machines, decision tree (random forest) induction systems, to name a few. Machine learning classifiers are designed to optimise global quantities such as accuracy without considering the intrinsic data characteristics such as class distribution. As a result, these algorithms ignore the minority class samples while accurately classifying the majority class during learning. Data imbalanced problems occur in many practical domains, and it is often handled using either data level approaches such as resampling, algorithm level approaches such as cost sensitive-based approaches or hybrid methods. Algorithm level approaches were reported to perform better than the data level methods [16], but they do not have the flexibility offered by resampling methods due to the difficulty of knowing the cost for each class. In particular, data resampling involve generating new samples or removing some existing samples. At the same time, the cost-sensitive approach tries to modify the learning algorithm or create new ones to respond favourably to both classes during learning. Ensemble learning methods are also increasingly considered as solutions to imbalanced problems[17]. The ensemble learning approaches are designed to either modify the learning algorithm (i.e. embedding a cost-sensitive strategy in the learning process) or use a data-level method (i.e. resampling the data) before the training stage of each weak classifier [18][19].

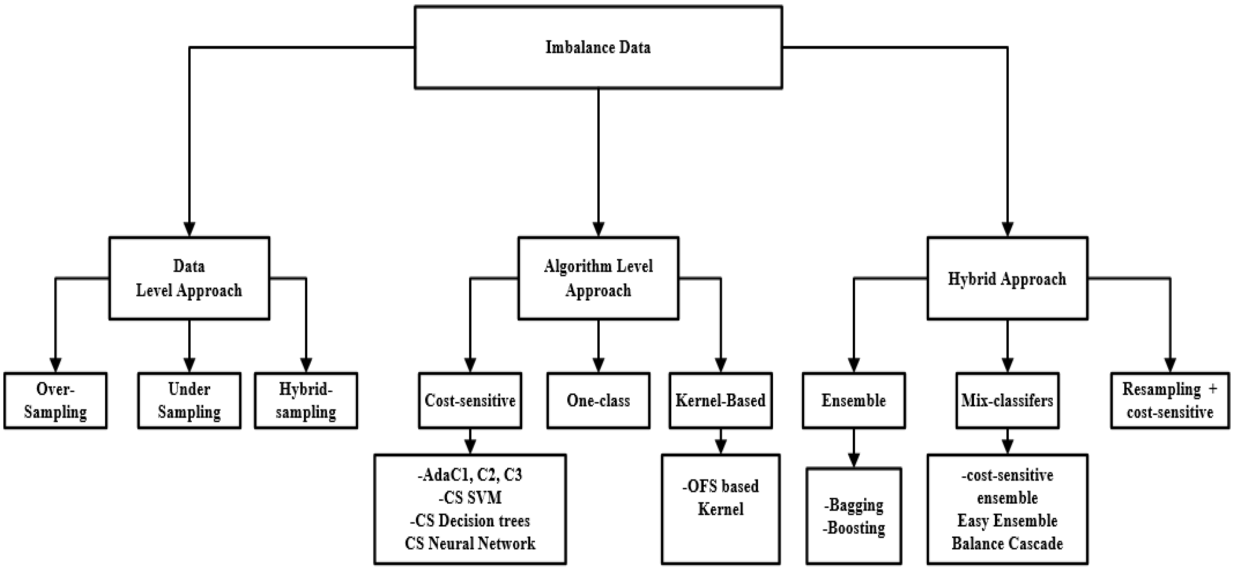


Figure 3- 2 Showing the three categories of the State-of-the-art approach of the handling imbalance problem

Several research approaches have been conducted to solve the imbalanced classification problem or rare failure prediction. Alberto et al. [12], Haixiang et al. [13], and Elrahman et al.[14] provided a detailed review of imbalanced learning. The Imbalanced classification problem can be grouped into three main categories: the data level, the algorithm level, and the hybrid approach, as seen in Figure 3-2. The data level approach involves resampling the dataset before presenting it as an input to the learning algorithm. The algorithm level approach tries to modify the traditional machine learning algorithm to respond favourably to both classes during learning [15]. The hybrid process involves combining two or more of either data-level or algorithm level techniques to achieve better performance.

3.1.2 Data level approach:- Resampling techniques

The open literature shows many studies focused on the resampling techniques to handle the imbalanced problem in different types of datasets. Those studies have empirically proven that balancing the class distribution before training is usually a useful solution [13][20]. The resampling techniques are categorised into three main groups the oversampling, undersampling, and hybrid. The data oversampling involve creating more examples either by replicating existing ones or creating new ones to balance the original dataset. The data undersampling involves eliminating examples from the majority class in order to balance with the minority class. The hybrid is the combination of oversampling and undersampling. Within these methods, the simplest techniques are random oversampling and random undersampling. One of the major disadvantages of random undersampling is that it can result in the omission of potentially useful instances that could be useful in the training process, such as in determining decision boundaries in support vector machine (SVM). Also, several authors have argued that random oversampling can increase the likelihood of overfitting problems because it replicates exact copies of the existing examples [13].

The challenges mentioned above have led to several proposed methods. Among them is the Synthetic Minority oversampling Technique (SMOTE). The (SMOTE) [21] has been developed to mitigate overfitting in random oversampling by taking a subset of data from the minority class as an example and then creating new synthetic similar instances. With this technique, minority class examples are oversampled by taking each instance and introducing an artificial instance along the line segment joining all or any **n** number of minority class nearest neighbours. The number of n nearest neighbours are randomly chosen depending on the number of oversampling needed.

However, SMOTE has the drawback of overgeneralisation in creating the synthetic instances, not considering neighbouring examples from other classes when generating synthetic samples. The SMOTE approach can crate overlapping of classes and can also introduce additional noise into the training data. SMOTE is also not effective in high dimensional data, as Lusa et al. [73] argued. Many solutions have been proposed to correct the drawbacks of SMOTE [67,74,75]. The majority of the novel solutions are specific to either the application domain or dataset in question, as presented by Alberto et al. [22].

Undersampling approaches are majorly based on data cleaning techniques where some instances are dropped following defined criteria. Examples of strategy in this area include the edited nearest neighbour rules (ENN) [23], where the examples that differ from two of its three neighbours are removed from the majority class. The NearMiss-2 method [24] selects majority class instances whose average distance to the three farthest minority class samples is the smallest [12]. Other approaches try to remove majority class instances that are far from the decision boundary [24]. Support Vector Machine (SVM) can also be used to discard irrelevant instances from the majority class. All the undersampling methods mentioned are prone to discarding useful information that can help in the learning process. Combining imbalanced problems with class overlapping makes the situation more complex and difficult to solve than handling it independently. A study has shown that identifying the overlapping region in the dataspace and dealing with those instances can make the data linearly separable [25]. The idea has been implemented in a [26] where SMOTE is used with Tomek link also CNN with Tomek link [27]. Other studies have investigated the effect of class overlapping on the machine learning classifier. The example is seen in Prati RC et al.[28]. They show that class overlapping aggravates the problem of imbalance and degrades the classifier's performance. In handling class overlapping problems, three stages are involved. Identifying the ambiguous or overlapping regions, managing the instances that belong to the region, and finally, the training or learning phase.

One of the major challenges in dealing with class overlapping is identifying the overlapping region. However, many solutions to identifying overlap regions in the dataset have shown success to some extent in partially imbalanced or balanced datasets. For example, the K-nearest neighbour based approach to identify overlapping areas on the data space has been proposed by Tang et al. [29]. The authors proposed a multi-model classifier known as DR-SVM, combining an SVM classifier with a kNN algorithm under a rough set technique. Trappenberg et al.[30] also proposed a strategy for identifying the overlapping regions in pattern classification. Similarly, fuzzy set representation of the concept that incorporates overlap information in the fuzzy classifiers is shown by Sofia et al. [29]. In addition, a one-class classification algorithm known as SVDD was used by Haitao et al. [31] to capture overlapping regions in an imbalanced dataset. Once the overlapping region is identified, the next step is managing data in that region. Haitao et al. [31] proposed that data with the overlapping region can be modelled using three ways. Discarding, merging and separating. The discarding involves ignoring data in the overlapping region learning from other data areas without overlap. An example of this approach is seen in SMOTE with Tomek link [26], where discarding is used to improve classifier performance in bioinformatics. In contrast, discarding schemes is successful in the balanced or slightly imbalanced dataset with enough training data in both classes. However, it will be impractical to apply that in data with absolute rarity or extremely imbalanced. Merging approaches involves merging the data in the overlap region as one class and then built a two-tier classification model with the upper-tier focusing on the whole dataset with an additional class and the lower-tier on the overlapping region [25]. A separating scheme involves separating that data into overlapping and non-overlapping and then treating each data subset separately to build the model. Tang et al. [29] proposed a multi-model classifier known as DR-SVM, combining an SVM classifier with a kNN algorithm under a rough set technique. The KNN is used to extract the overlapping region. Then, two SVM are used to train the two subsets of the resulting data at the end of the learning process; the classification result will show whether the patterns lie in an overlap region. The classification of test examples belonging to the overlap or non-overlapping region depends majorly on the aim of the application domain. The application of this approach will require a domain expert to be determined the class of the test examples. The drawback of this approach is that it can not be used in intelligent systems where domain expert input is not required to make decisions for the next task. All the approaches mentioned above either consider the overlapping data as noise and drop it during learning to increase the confidence level of the model.

In a situation where extremely rare classes exist, handling the class overlapping in an imbalanced distribution is more challenging. In the case of the ACMS dataset, class overlapping occurs due to the fact that there is an infrequent number of components replacement (extremely rare minority) confirmed by data visualisation. Applying the approaches mentioned above is not feasible because the extremely rare examples could be ignored as noise, and also, intelligence aircraft maintenance systems are required to decide with absolute certainty due to the time-critical nature of the asset rather than waiting for domain experts intervention. Therefore, the particularities of our data (Heterogeneous in nature containing symbolic sequences, numeric time-series, categorical variables and unstructured text. It is a non-trivial task to translate free-text log messages into meaningful features) limit us from using an out-of-the shelf approach. A newer approach is required to handle class overlapping in an ACMS dataset.

3.1.3 Algorithm level techniques: Cost-Sensitive Approaches

The algorithm level approach tackles the imbalanced learning problem by altering the learning algorithm or creating new ones to respond favourably to both classes during learning [15]. Cost-sensitive learning is an algorithm-level approach. The cost-sensitive method is explored by defining the cost of misclassification for each class. Determining the cost of misclassification is challenging in the traditional classification algorithms (such as support vector machines, decision trees, and more) because such algorithms presume that all classification errors carry the same cost. As a result, the loss function ignores the data distribution and focuses on minimising global values such as the error rate. Hence, majority-class examples are correctly classified, while minority-class samples are incorrectly classified. Indeed this solution can be accepted if the aim is to simply maximise the accuracy (that is, minimising the error rate) without minding the misclassification cost. However, suppose the rare examples are more important to classify, such as in the predictive maintenance domain (predicting failure is critical). In that case, more robust methods are required to handle the learning process. Moreso, in cases where the minority class is extremely rare, the algorithms can treat the examples as outliers of the majority and ignore them during learning [32]. In such a situation, the learning algorithm will end up creating a trivial model that will classify every example as a majority. Therefore, cost-sensitive learning takes into consideration the different costs that vary by type of classification (true-positive, true-negative, false-positive, false-negative) across all samples with respect to different classes [16].

A cost matrix is used in cost-sensitive learning to assign class penalties to each class, which shifts class boundaries to reduce biases induced by class imbalance [33]. The cost matrix controls misclassification problems during the learning process. The most common approach used is increasing the cost of the minority class samples. The classifier will give more importance to the minority class during learning, decreasing the likelihood of misclassifying the minority samples as a majority [33].

Three general main cost-sensitive approaches have been proposed to deal with imbalanced class problems, the direct methods, the meta-learning methods, and the re-weighing methods. The direct techniques involve utilising or introducing misclassification cost directly into the machine learning algorithm [12]. Examples of such approaches can be seen in algorithms such as decision tree induction, where the misclassification costs are minimised through tree-building strategies [34]. The cost information is either used to split the data [35] or determine the subtree's pruning condition [36]. Also, some methods include misclassification cost in the loss function; the example is seen in Turney [37] hybrid genetic decision tree induction algorithm for cost-sensitive applications. The re-weighting cost-sensitive methods involve adjusting samples importance by assigning weights of different values to them. The most commonly used approach is re-weighting samples inversely proportional to the class size [38], the square root of class frequency [39], and the term frequency factor in supervised term-weighting schemes for text Classification [40]. Instead of using the number of classes heuristically Cui et al.[41] approach it differently, and they proposed a re-weighting scheme that uses the effective number of samples for each class to re-balance the loss. Similarly, Yu-Xiong et al. [42] proposed a transfer learning approach for a skewed distribution dataset where knowledge from the data-rich classes in the head of the distribution is transferred to the data-poor classes in the tail. The model is designed to learn from the tail of a class distribution where minority data is available. These methods only focus on the class distribution at the global level, where the fixed weight is assigned to all samples in each class. However, not all samples play the same role in determining the model parameters (descriptive measures of an entire dataset that are used to generate distribution) [33][43]. That is, some samples have more contribution to determining the decision boundaries of the model than others. Hence, methods to re-weight is sample according to its effect on determining the models' decision boundary are required for effective learning.

Many studies have been conducted recently to deal with sample-based re-weighting by the use of loss function. Qi et al.[44] developed a Class Rectification Loss (CRL) regularising algorithm for addressing class imbalanced problems. Tsung-Yi et al.[45] By redesigning the traditional cross-entropy loss, they developed a loss function known as focal loss (FL). It reduces the significance of the loss ascribed to well-classified cases. Tomasz et al. [46] proposed a method based on training a separate linear SVM classifier for every example in the training dataset to handle imbalances. These approaches basically re-weights the training data by down-weighting the majority class and up-weighting the minority class samples.

The re-weighting process may lead to training the whole dataset, which can be time-consuming, especially in the case of deep neural networks that are capable of memorising the complete dataset. The Deep Neural Networks (DNN) are trained to find complex structures in a dataset by using a backpropagation algorithm. The algorithm calculates errors made by the model during training, and the models' weights are updated in proportion to the error. The drawback of this learning method is that examples from both classes are treated the same. In that situation where the data is imbalanced, the model will be adapted more to the majority class than the minority class, which means the model will be overfitted to the majority class samples, which are located at the overlapping region of the dataset affecting the performance of the models.

The meta-learning method involves learning from the output of other machine learning algorithms. Most commonly, this means the combination of pre-processing techniques for training data or a combination of predictions from other machine learning algorithms, i.e. postprocessing of output from other algorithms without altering or modifying the original algorithm [47][48]. Meta-learning provides an enhanced learning paradigm where models can gain experience after many learning episodes, covering the distribution of related patterns in the dataset and using the experience to improve model performance [47]. Meta-learning methods are further divided into two, thresholding and sampling. Thresholding is based on basic decision theory that assigns instances to a class based on the minimum expected cost. An example of thresholding can be seen in the decision tree algorithm for the binary classification problem. A class label of a leaf node is determined based on the majority class of the training sample that reaches the leaf nodes (if most of the training samples at leaf are positive, the label is assigned as positive otherwise negative) [49]. Whereas cost-sensitive decision tree algorithms assign a class label to a node that minimises the classification cost [22][50]. On the other hand, the sampling meta-learning approach is based on modifying the class distribution based on cost metrics, and this is mainly achieved through non-heuristic resampling methods.

Recently, cost-sensitive meta-learning approaches have been developed to enhance the performance of re-weighting and direct cost methods [51][52]. Shu et al.[53] proposes a meta-learning process known as Meta-Weight-Net to lean the weighting function. Their approach is designed to learn an explicit weighting function from data adaptively instead of manually pre-specifying the weighting function. Also, Olowookere et al. [54] proposed a cost-sensitive meta-learning ensemble approach for detecting credit card fraud, Kulluk et al. [55] proposed a classifier based on Cost-sensitive meta-learning and re[56]sampling techniques, Liu et al.[57] also suggested and resampling method by integrating meta-learning ensemble methods. Similarly, Ren et al. [58] proposed a meta-sampler with a balanced meta-softmax function as an extension of softmax function [59] for long-tailed Visual Recognition. Although these methods have some performance improvement levels, they are challenging to apply in real-life cases. For instance, Meta-Weight-Net [53] requires additional balanced distributed data for training. Also, meta-sampler [58], the learning process is computationally expensive. Therefore, to achieve good performance, learning from imbalanced data using cost-sensitive methods is to design a loss function that does not require a hyperparameter or specially designed architecture that integrates data re-sampling with cost-sensitive.

Another method for handling an imbalanced dataset is one-class classification. It is a method for finding anomalous data points compared to known-class examples, and it can help with difficulties like severely imbalanced datasets [56]. In other words, one-class classification methods focus on and analyse only one class, which is usually the one of interest. In the case of an imbalanced classification problem, the labelled examples of the positive class(es) are either unavailable or insufficient to train a normal machine classifier [56]. Under some conditions, such as multi-modality of the domain space, one-class approaches to solving the classification problem may actually be superior to discriminative (two-class) approaches, such as decision trees or Neural Networks[60][61]. Many one-class solutions to imbalanced datasets have been proposed in the literature. Seliya et al.[56] provided a general literature review on one-class classification methods and their potential applications. Some of the one-class methods focus on the intrinsic characteristic data, and their interrelationship to the extreme imbalanced classification problem are. For example, to capture the overlapping regions in real-life datasets, Xiong et al.[31] employed the one-class classification algorithm Support Vector Data Description (SVDD). Raskutti et al. [62] show that one class learning is most useful when applied to extremely imbalanced sets of data with a high dimensional and noisy feature space. They believe that the one-class strategy is similar to robust feature selection approaches but that it is more practical because feature selection can be costly to implement. One-class learning methods for time-series data are seen in Yamaguchi et al. I. [63] suggested a one-class learning time-series shapelets approach known as OCLTS. The OCLTS uses a stochastic sub-gradient descent approach to efficiently and simultaneously optimise shapelets and a non-linear classifier based on a one-class support vector machine. Experimental findings demonstrate the method's usefulness for interpretability and imbalanced binary classification. In Mauceri et al. I. [64] time-series are represented as vectors of dissimilarities derived from a set of prototypes. They analyse a Cartesian product of Twelve dissimilarity measures and Eight prototype techniques using this method (strategies to select prototypes). The dissimilarity-based representations are classified using a one-class nearest neighbour classifier (DBR). They claim to be the first to do a full one-class classification experiment using the literature's largest library of time series data sets. Finally, one-class classification methods for time-series concentrating on an extremely imbalanced dataset with class overlapping and minor class disjunct are lacking in the open literature.

3.1.4 Ensemble Learning Techniques for Imbalanced ACMS Data

Ensemble learning is a methodology where multiple machine learning models are trained to solve the same problem, and the output of the learner is combined to get improved performance. The method tries to improve the machine learning classifier's performance by combining the decision of other classifiers, known as weak learners [34][12]. The major course of error in machine learning algorithms is the presence of noise, variance, and bias in the dataset [65][50]. Ensemble classifiers are built to minimise these factors, which improves machine learning algorithms' stability and learning performance. A study by Zhou et al. [65] shows a broad overview of why and how ensemble learning improves prediction performance. The two most basic qualities expected of a model are a low bias and a low variance, which frequently fluctuate in opposite ways. Indeed, the model is required to have enough degrees of freedom to resolve the underlying complexity of the data it is working with, but not too many degrees of freedom to avoid high variance and be more robust. This is the well-known tradeoff between bias and variance.

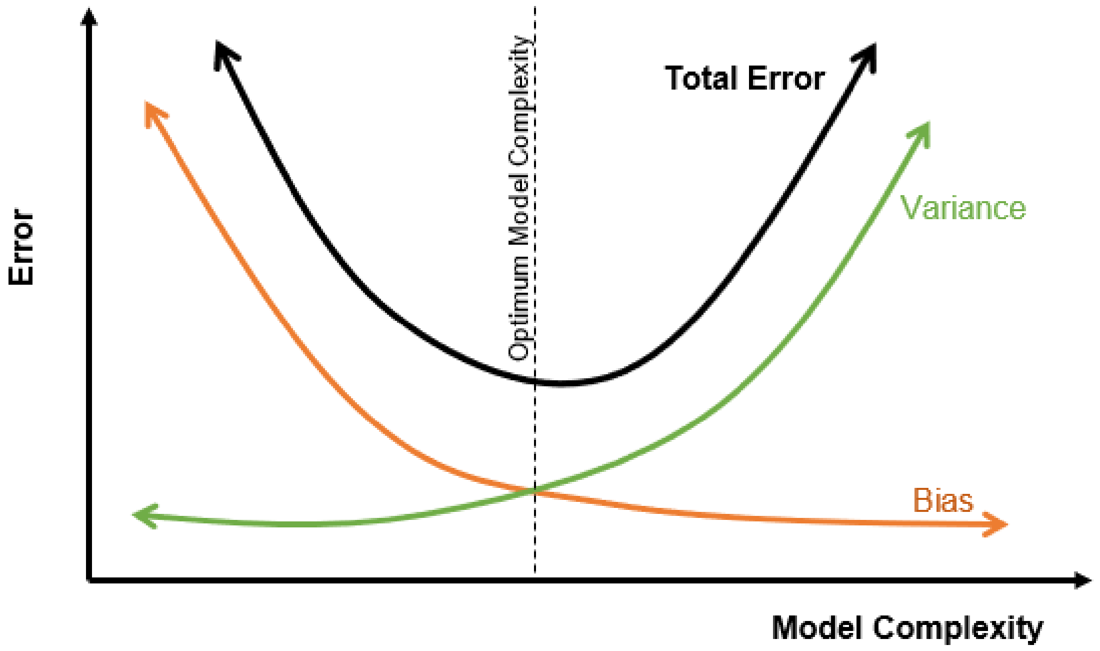


Figure 3- 3 Bias-Variance Tradeoff

Most of the time, in ensemble learning, the weak learners don't perform well by themselves either because they have a high bias or high variance. Therefore, the idea of the ensemble is to try reducing the variance and bias at the same time by combining multiple weak classifies to create a strong one for enhancing performance. The ensemble learning strategy can be constructed based on the flowing approach, boosting, bagging and stacking learning structures. The Bagging (bootstrap aggregating) methods involve training homogeneous weak classifies in parallel and independently and then combining them by averaging process [34]. The bagging implementation that is bootstrap aggregating [66] can be seen in SMOTEBagging [67]. The boosting method involves training the homogeneous weak classifies sequentially in an adaptive way and combining them following a deterministic strategy.[34]. The implementation of boosting learning can be found in AdaBoost [68], SMOTEBoost [69]. The stacking approach consists in training the homogeneous weak classifies in parallel and combining them using a meta-learning model to output the prediction of the bases learner on the different weak models [50].

In recent years diverse ensemble learning strategies has risen as a possible solution to the data imbalanced problems [70]. For instance, Galar et al. [71] provided a broad overview of different combinations of multiple classifiers to improve predictive accuracy using an imbalanced dataset. López et al.[12] studied about insight into the characteristics of the imbalanced dataset scenario in classification. The hybrid-ensemble approach is another ensemble learning-based method that shows a more promising impact on imbalanced classification. The hybrid process involves incorporating the resampling or cost-sensitivity in either the weak classifier or base classifier levels. For example, an extensive study about hybrid classifies is seen in Wozniak [72], Galar et al.[73] develop an ensemble-based algorithm known as EUSBoost based on RUSBoost, which combines random undersampling with Boosting. Also, Krawczyk et al. [18] use a cost-sensitive ensemble-based decision tree algorithm to classify highly imbalanced datasets. López et al.[74] provided a general analysis of pre-processing with cost-sensitive methods. Liu et al. [75] integrate AdaBoost, and an ensemble learning approach is used to detect overlapping data. The study successfully in handling overlapping class problems, which is understandable when the data is balanced. However, it was not clear in the case where one class is extremely imbalanced combined with overlapping. Although, much research on tackling the imbalanced data problem using ensemble-based learning has been provided in the open literature [70]. The majority of them focus on either imbalanced data without minding other data complexities such as the overlapping or small class disjunct or focus on one of the complexities alone. Therefore, the ensemble methods still face challenges in tackling imbalanced datasets with the combination of extremely rare minority and overlapping issues. The open literature lacks an extensive study that uses ensemble learning to address extreme rarity, class overlapping, and class disjunct especially using a system log dataset. Therefore, the arising research question is, how can an architecture of ensemble classifiers be contructed for tackling extremely imbalanced datasets taking into account class overlaping and small disjunct problems. Usually, the number of weak learners is selected arbitrarily, which can result in redundancy for similar classifies [70]. For example, relating the size of weak learners to the data complexities such as reducing bias and variance in the extremely imbalanced dataset.

3.1.5 Deep Learning Techniques for Imbalanced Datasets

Deep learning is a branch of machine learning consisting of numerous processing layers that learn data representations at multiple levels of abstraction using artificial neural networks (ANN). Deep learning models have greatly improved the state-of-the-art performance of models in many domains, such as large-scale data processing, image detection, and time series analysis, to name a few [7]. The success has been attributed to an increase in the availability of data, hardware, and software improvements and many breakthroughs in algorithm development that speed up training and other data generalizations [16]. Despite the advances, little work has been done to investigate the effect of extremely imbalanced, class overlapping, and small class disjunct on the deep neural network architectures. Many researchers have agreed that the subject of imbalanced data with deep learning is understudied [76–79]. For resolving data imbalanced problems in predictive modelling, deep learning technologies can be combined with either data level or algorithm level solutions. In deep learning, the ANNs are trained to find complex structures in a dataset by using a backpropagation algorithm. The algorithm calculates errors made by the model during training, and the models' weights are updated in proportion to the error. The drawback of this learning method is that examples from both classes are treated the same. In a situation where the data is imbalanced, the model will be adapted more to the majority class than the minority class, which can affect the performance of the models [16]. The majority of the deep learning methods for imbalanced classification have depended on integrating either data resampling or cost-sensitive methods into the deep learning process. For instance, Hensman et al. [80] use random oversampling techniques to balance the data then train the balanced data using CNN. Similarly, Lee et al.[81] uses Random undersampling to balance the dataset for the purpose of pretraining CNN. The use of dynamic sampling to adjust the sampling rate according to the class size for training CNN was proposed by Pouyanfar et al. [82]. Buda et al. [79] investigate the effect of random oversampling, random undersampling and two-face learning across using several imbalanced datasets on deep neural networks. The literature review [16][83] reveals that most of the proposed deep learning resampling approaches for imbalanced problems use image datasets and CNN architecture. The need to Investigate the effect of imbalanced on other deep learning architectures and to use time-series is still lacking.

On the other hand, some studies have focused on solving the challenge of imbalanced classification using cost-sensitive methods, which involves modifying the deep learning process to favour both classes during model training. For example, Khan SH et al. [84] proposed a cost-sensitive deep neural network that can automatically learn robust feature representations for both the majority and minority classes. Also, Zhang et al. [32] propose cost-sensitive deep belief networks, and Wang H et al. [85] propose a cost-sensitive deep learning approach to predict hospital readmission. Also, the use of loss function to control biases has been shown in Wang S et al. [6]. The authors proposed a novel loss function called mean false error and its improved version mean squared false error for learning from an imbalanced dataset. Similarly, a new loss function called Focal loss was proposed by Lin et al. [45] for dense object detection in image classification. The focal loss was proposed to specifically handle the challenge of extreme data imbalances commonly faced in object detection problems, where the foreground samples usually outnumber the background samples. Normally, this type of problem is mostly solved using the one-stage detection approach or two-stage detection. The two-stage detection usually performance at the cost of computation time as compared to one-stage. In Lin et al.[45] study, they focused on determining how the one-stage approach with fast computation time can achieve a state-of-the-art performance compared to the two-stage. Their study discovered that the main cause of performance degradation in one-stage detection is the imbalanced data problem. The overwhelming background samples create imbalance, causing the majority class to account for most of the overall loss. To address that challenge, Lin et al. [8]. Proposed a loss function known as the focal loss (FL) which was derived from a normal binary cross-entropy loss. The FL is expressed as follows;

Focal Loss $\mathrm{FL}(p_{,t}) = -(1 - (p_t))^{\gamma} log_{10} (p_t)$ (3- 1)

The new FL tries to reduce the impact that the majority of samples have on the loss by multiplying the cross-entropy loss with a modulating factor $-(1 - (p_t))^{\gamma}$ Where the hyperparameter γ ≥ 0 adjusts the learning rate, the negative samples are downweighed. Their implementation shows that using one-stage detection with focal loss by selecting the right learning rate outperformed the two-stage approach. The implantation method was only compared with cross-entropy and tested for imbalance problems in objection detection. The focal loss was later tested in image classification by K Nemoto et al. [86]. The authors used CNN architecture then compare the performance of focal loss to cross-entropy loss for image classification. The open literature lacks a study investigating the focal loss's effectiveness on time-series systems log-based datasets, particularly for datasets such as the log-based ACMS.

3.1.6 Deep Reinforcement Learning for imbalanced dataset (ACMS)

Deep reinforcement learning (DRL), which combines deep neural networks with reinforcement learning to produce advanced solutions, is garnering more academic attention and delivering state-of-the-art solutions, particularly for performance optimization [87]. For example, combining deep learning and reinforcement learning has resulted in the development of a new method known as the deep Q-network (DQN) [88–90]. DRL has made the use of reinforcement learning appealing in a variety of domains. The development of predictive maintenance models for complex systems such as aircraft is one such domain that can benefit from DRL. The algorithm level method or the data level technique can be used as a hybrid to DRL for imbalanced classification in predictive modelling.

Deep reinforcement learning has recently shown promising results for data classification since it can aid classifiers in learning crucial characteristics or selecting good instances from heterogeneous data [91]. Feng et al. [92] constructed a deep reinforcement learning model for relation classification at the sentence level from noisy data. The learning process is divided into instance selectors and relational classifiers. The instance selector is an agent that selects high-quality sentences from input, whereas the relational classifier learns from previous data and rewards the instance selector. Finally, the model acquires a higher-quality data set as well as a more effective classifier. Martinez et al. [93] propose a reinforcement learning framework for early classification in time-series data. The method introduces a set of states and actions and defines a reward function that aims to find a compromise between earliness and classification accuracy. Hashemi et al. [94] presented an ensemble pruning strategy that used reinforcement learning to choose the best sub-classifiers. However, because selecting classifiers was inefficient when there were many sub-classifiers, this strategy was only suitable for traditional small datasets[89]. Lin et al. [89] formulated an imbalanced classification problem as a Makov sequential decision-making process that uses agents as a classifier interacting with the environment to obtain an optimal policy. However, the process created a high time complexity due to the interaction between agent and environment.

The available literature is lacking in many works on imbalanced classification using deep reinforcement learning. Also, the ACMS dataset lacks methodologies that have transformed DRL techniques for rare failure prediction modelling. As a result, a thorough investigation of the applicability of deep reinforcement learning for exceedingly rare event prediction in the context of predictive maintenance modelling is necessary.

3.1.7 Imbalance learning: Summary of Key Literature Findings

Table 3- 1 Key Literature Findings

Key Findings			
Stage	Techniques	Strength	Weakness
Data Level	Sampling (over-sampling, Under-sampling and hybrid)	It is suitable for large datasets and slightly imbalanced problems. It gives a better approach to detecting minority class. It is easy to implement.	It is prone to changing the original structure of the dataset, which can impact models' performance. Random oversampling can cause an overfitting problem. Under-sampling can reduce informative data points, and it discards potential useful data.

			Oversampling increases computational time.
	Feature Engineering: (Wrapper, Embedded, and Filter) [95–98]	Reduce susceptibility, overfitting, storage memory, and processing. It reduces computational time and cost. Suitable for both time series, discrete and continuous datasets. It is ideal for a dataset with a high imbalanced ratio. Suitable for high dimensional datasets.	Measure features independently without considering Interaction between all features
	Dimension reduction (PCA, LDA and Autoencoder) [99–103]	Produce useful features for learning. Measure features dependently, considering Interaction between all features.	Increases computation time.
Algorithm Level	Cost-Sensitive Learning	Produce good results for the minority class.	Misclassification costs are often not known.
	Ensemble technique with Iteration	Produce a strong learner by combining two or more weak learners.	The high cost of computation.

3.1.8 Performance Metrics in imbalanced learning domains

This section contains the performance metrics that have been used throughout the thesis. Evolution criteria are critical when evaluating the classification performance of a machine learning algorithm, especially when the data is extremely imbalanced.

3.1.8.1 Classification Matrices

Correct classification and misclassification results can be shown using a confusion matrix in a binary classification problem, as seen in Table 3-2.

Table 3- 2 Confusion Matrix

	Actual Positives	Actual Negatives	TP: True Positive
Predicted Positives	TP ($C_{1,1}$)	FP ($C_{1,-1}$)	TN: True Negative
Predicted Negatives	FN ($C_{-1,1}$)	TN ($C_{-1,-1}$)	FP: False Positives FN: False Negative

When an example from the majority class is misclassified as an example from the minority class, a false-positive arises. False-positive is less serious than false-negative, which occurs when a member of the minority group is mistakenly labelled as a member of the majority group. In this study, the term "false-negative" refers to misclassifying a components fault as "healthy," which is particularly risky because it could result in equipment damage. Similarly, false-positive means misclassifying a healthy component as faulty; It is possible that the cost of maintenance checks will rise as a result of this. One of the most significant challenges in imbalanced classification is the cost of misclassification, which is difficult to accurately define. True-positives and true-negatives are the correct classifications of positives and negatives, respectively.

As seen in Table 3-2 for classification models, confusion matrices are often used to measure effectiveness in the validation of models. In this study, component failures are considered a positive class, while non-failure is considered a negative class.

Definition of formulae used in this study.

Patterns with component Failure = Positives

Patterns without component Failures = Negatives

True Positives (TP) = patterns with components failures that have been classified as a failure.

True Negatives (TN) = patterns without component failure who have been classified as non-failure

False Positives (FP) = patterns with components failure who have been classified as non-failure

False Negatives (FN) = patterns without components failure who have been classified as failures

Common metrics extracted from these are:

The true-positive rate (TPR), also known as Sensitivity, measures the proportion of components with failure who have been classified as component failures.

$$\text{TPR/sensitivity} = \frac{TP}{TP+\mathrm{FN}} \tag{3- 2}$$

The false-negative rate (TNR), also known as Specificity, measures the proportion of components without failure that has been classified as non-failure components.

$$\text{TNR/Specificity} = \frac{TN}{TN+FP} \tag{3- 3}$$

The false-negative (FNR) measures the proportion of patterns without components failure who have been classified as failures.

$$\text{FNR} = \frac{FN}{TP+FN} \tag{3- 4}$$

False Positive rate (FPR) measures the proportion of patterns with components failure that has been classified as non-failure.

$$\text{FPR} = \frac{FP}{FP+FN} \tag{3- 5}$$

A false-positive arises when an example (pattern) from the minority class is misclassified as an example from the majority class. False-negative is less serious than false-positive when patterns with component failures are considered positives (minority class) and patterns without component failures are considered negatives. The term "false-positive" is used in this study to describe misclassifying a malfunctioning component as "healthy," which is particularly dangerous because it could cause equipment damage. Similarly, a false negative involves misclassifying a working component as faulty; as a result, the extra cost of maintenance checks may increase.

The performance measurements are often used to compare several models using a ROC plot. In the case of failure prediction of an aircraft component, sensitivity is the model's ability to correctly predict failure, leading to component replacement (probability of positive prediction given that the failure results in component replacement). Model specificity relates to the model's ability to correctly predict non-failure, resulting in no replacement (probability of negative prediction given that no failure occurs).

Typically, accuracy is regarded as the most major parameter for assessing a classifier's performance. However, using accuracy to measure performance in extreme imbalanced issues can be misleading since, in order to attain high overall accuracy, classifiers would be biased towards the majority class. For example, a classifier that achieves a 90% accuracy in a dataset with a 5% imbalance ratio is not accurate if it labels all cases as negative. Accuracy can be represented as

$$\text{Accuracy} = \frac{\boldsymbol{TP}+\boldsymbol{TN}}{\boldsymbol{TP}+\mathbf{FN}+\boldsymbol{FP}+\boldsymbol{TN}} \tag{3- 6}$$

Some alternative metrics are created using the confusion matrix to measure the classifiers' performance more precisely in the presence of an excessively imbalanced situation, taking into consideration the class distribution.

True Positive Rate (TPR) Measure the percentage of positive examples that are correctly classified, while True Negative Rate (TNR) Measures the percentage of negative examples that are correctly classified.

False Positive Rate (FPR): Measures the percentage of negative examples that are misclassified which is represented as.

$$\text{FPR} = \frac{FP}{FP+TN} \tag{3- 7}$$

False Negative Rate (FNR): Measures the percentage of positive examples that are misclassified, represented as.

$$\text{FNR} = \frac{FN}{TP+FN} \tag{3- 8}$$

Precision (p): is the measure of classifier exactness, the percentage of true positive predictions made by the classifier that is truly correct. So, low precision indicates a large number of False Positives, represented as.

$$\text{Precision (p)} = \frac{TP}{TP+FP} \tag{3- 9}$$

Recall (r) is the classifier completeness measure and is defined as the percentage of true positives that the classifier can correctly detect. So, low recall indicates many False Negatives, represented as.

$$\text{Recall (r)} = \frac{TP}{TP+FN} \tag{3- 10}$$

F1-Score or F-measure (F1): Measures the harmonic mean precision and recall represented as.

$$\text{F1:-}\ 2 * \frac{Precision*Recal}{Precision+Recall} \tag{3- 11}$$

The Geometric Mean (G-Mean) is a metric that measures the balance between classification performances on both the majority and minority classes. G-mean measures the root of the product of class-wise sensitivity; it attempts to maximise each class's accuracy and keeps the accuracy balanced. This measure is important in avoiding overfitting the negative class and underfitting the positive class. G-mean is represented as.

$$\text{G -mean} = \sqrt{\boldsymbol{p} * \boldsymbol{r}} \tag{3- 12}$$

Receiver Operating Characteristic Curve (ROC) Curves: ROC is a graphical representation that illustrates the classifier's diagnostic ability as the discriminant threshold is varied. An excellent model has an area under the curve AUC with a value near one, meaning the model has a good separability measure.

Assuming we have two classes, the positive and negative classes ROC curve of those classes' probability would be.

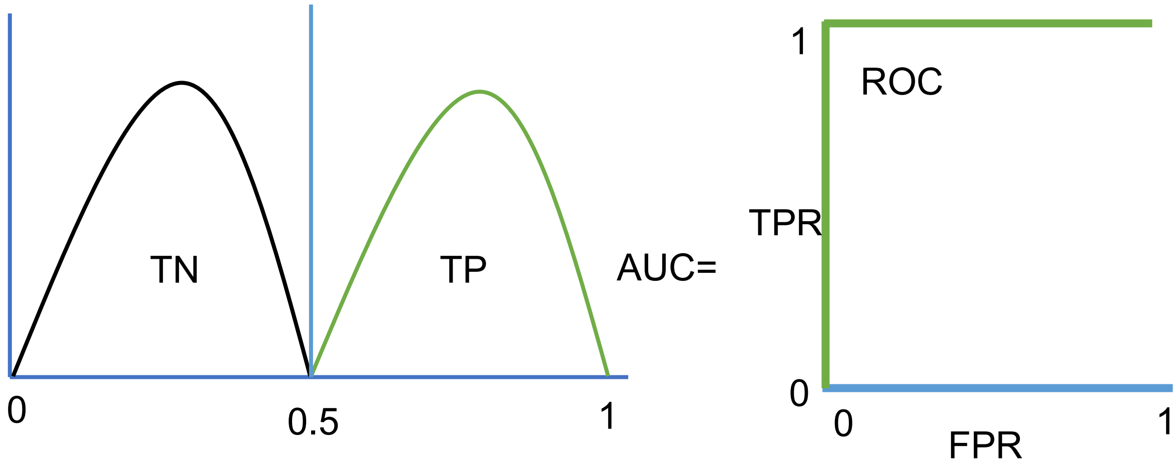


Figure 3- 4 ROC curve for an ideal situation

Figure 3-4 shows the ROC curve for an ideal situation. The green distribution curve represents the positive class (component failure), and the black distribution curve represent the negative class(non-failure). When the two curves do not overlap, the model has an ideal separability measure (the model can correctly distinguish between positive and negative classes).

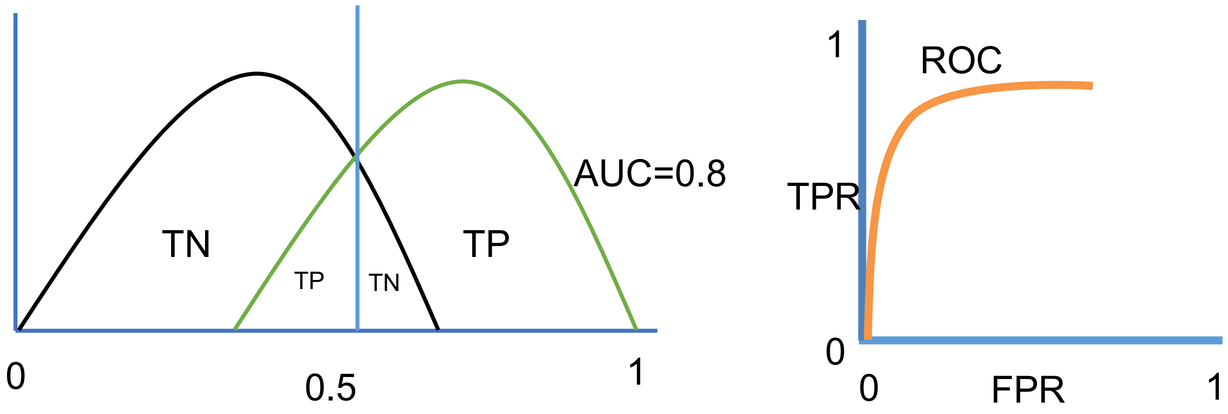


Figure 3- 5 ROC Curves showing overlap distributions with AUC=0.8

Figure 2-5 shows a situation when two distributions overlap. In this case, type 1 and type 2 errors will be introduced. Based on the value of the threshold, the error can be minimised or maximised. When AUC is 0.8, the model has an 80% chance of distinguishing between positive and negative. The model AUC = 0 is the worst separability measure (the model is reciprocating the classes, which means the model predicts a negative class as a positive class and vice versa). When AUC = 0.5, it means the model has no class separation capacity whatsoever.

3.2 The Big Data Analytics in Aerospace

The term 'Big Data' represents a new generation of technologies and architectures designed to extract insight from a large amount of data by allowing processing and analysis in real-time. Big data is characterised by volume velocity variety and veracity in some cases with large magnitude [104]. Data analytics is the process of inspecting, cleansing, transforming and modelling data using diverse techniques to discover useful information from data, providing actionable insight for support and decision-making [105][106]. The application of data analytics can be seen in many domains for different purposes, as shown in Figure 3-6. Examples of those domains are healthcare, banking and finance, engineering, aerospace, marine, to name a few. Big data analytics is used in the banking and finance industry to detect financial crimes and fraud, enhance risk management, and understand customer behaviour patterns [107][108]. In healthcare, data analytics is used to detect diseases and monitor patients' conditions [109]. In aerospace, big data analytics is changing the way airlines are doing business and enhancing system availability by reducing unscheduled maintenance. Badea et al. [104] and Xia BS et al. [105] studied the application of data analytics in different domains and pointed out that data analytics solutions are more domain specifics, meaning solutions from one domain cannot be directly applied to another.

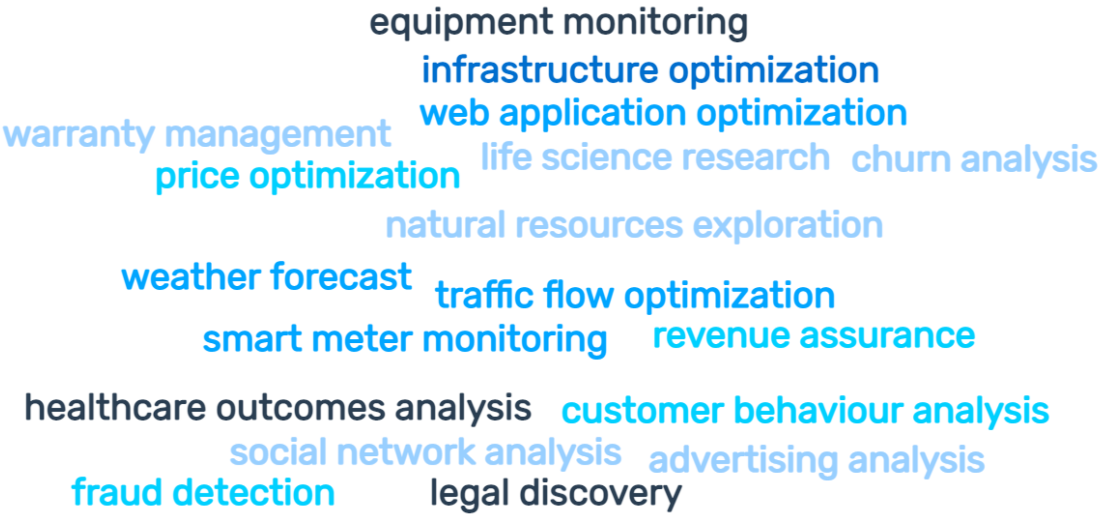


Figure 3- 6 Industrial applications of big data

The recent advancement in artificial intelligence (AI) technologies, such as applying the Internet of Things (IoT) in the manufacturing system, has produced a technology known as industry 4.0. These technologies are actualising the movement towards smart manufacturing (also known as digitisation manufacturing). OEMs embed even more sensors on the different aircraft components to monitor and record a larger number of aircraft system parameters that contribute to the large amount of data generated from modern aircraft. Big data analytics is increasingly becoming relevant in the aerospace industry, making it possible to move towards 'smart aircraft'. Smart aircraft technology is when aircraft components can communicate (components -to-Component’s communication) without humans' interference. Embedded devices can carry out edge analytics, and the output can automatically influence other systems working conditions. The component's interactions can enable the component to automatically adjust to working conditions based on an output from other components. An example can be seen as follows; A self-aware cabin system, which monitors the sensitivity of loads, overhead storage bins will immediately indicate when passengers stow luggage so that other passengers won't have to search around to find a free spot.

Aircraft smart seat systems can also report empty seats and show passengers who have not fastened their belts, trays, and legs. This will make boarding more manageable and help keep flights on schedule. The data generated by the smart systems can be used to retrain the system for performance improvement, such as making the aircraft more responsive to passengers demands. Many research has shown different approaches to implementing smart systems in a modern aircraft system. For instance, Daniel et al.[110] shows the architectural design and the conceptual framework for a smart maintenance decision system using big data analytics. Devish et al.[111] has reviewed some deep learning approaches for aircraft maintenance.

The power of intelligent maintenance systems lays in the historical dataset. According to Oliver Wyman's survey of the year, 2019 estimated that the global fleet of aircraft would generate more than 98 million terabytes of data by the year 2026 [112][113] (see Figure 3-7). The large volume of data collected and proper application of artificial intelligence technologies could significantly transform modern aircraft operations.

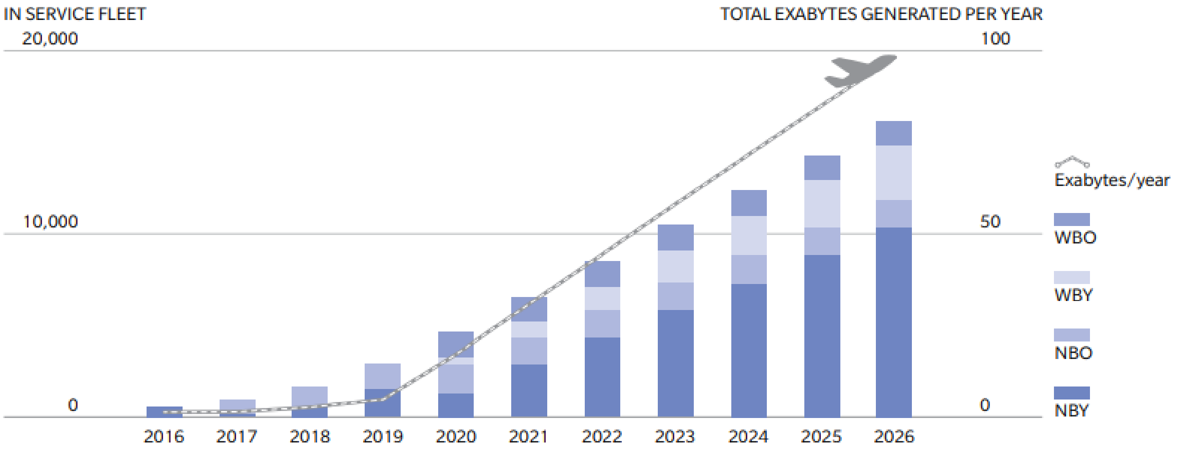


Exhibit 1: Data generated from projected global fleet

In 2026, the global fleet will generate 98 exabytes of data (That's 98 million terabytes or 98 billion gigabytes)

Source: Oliver Wyman Fleet & MRO Forecast, www.planestats.com/betterinsight

Figure 3- 7 Oliver Wyman's survey on the future of big data in aerospace [112].

3.2.1 Artificial Intelligence -AI

Artificial Intelligence (AI) is created by a computational study of how the human brain functions (how humans think and make decisions). The goal of AI is to build machines that act, work, and possibly think like humans. Data-based (Machine learning) and symbolic-based AI are two different types of AI (also known as symbolic learning). There are various sections of AI, as shown in Figure 3-8. Speech recognition (SR) is a field that attempts to replicate how humans communicate using language by listening and speaking. Natural Language Processing (NLP) is an area of artificial intelligence that allows robots to read and write text in the same way humans do. Computer vision gave robots the ability to see with their eyes and process information in the same way that humans do.

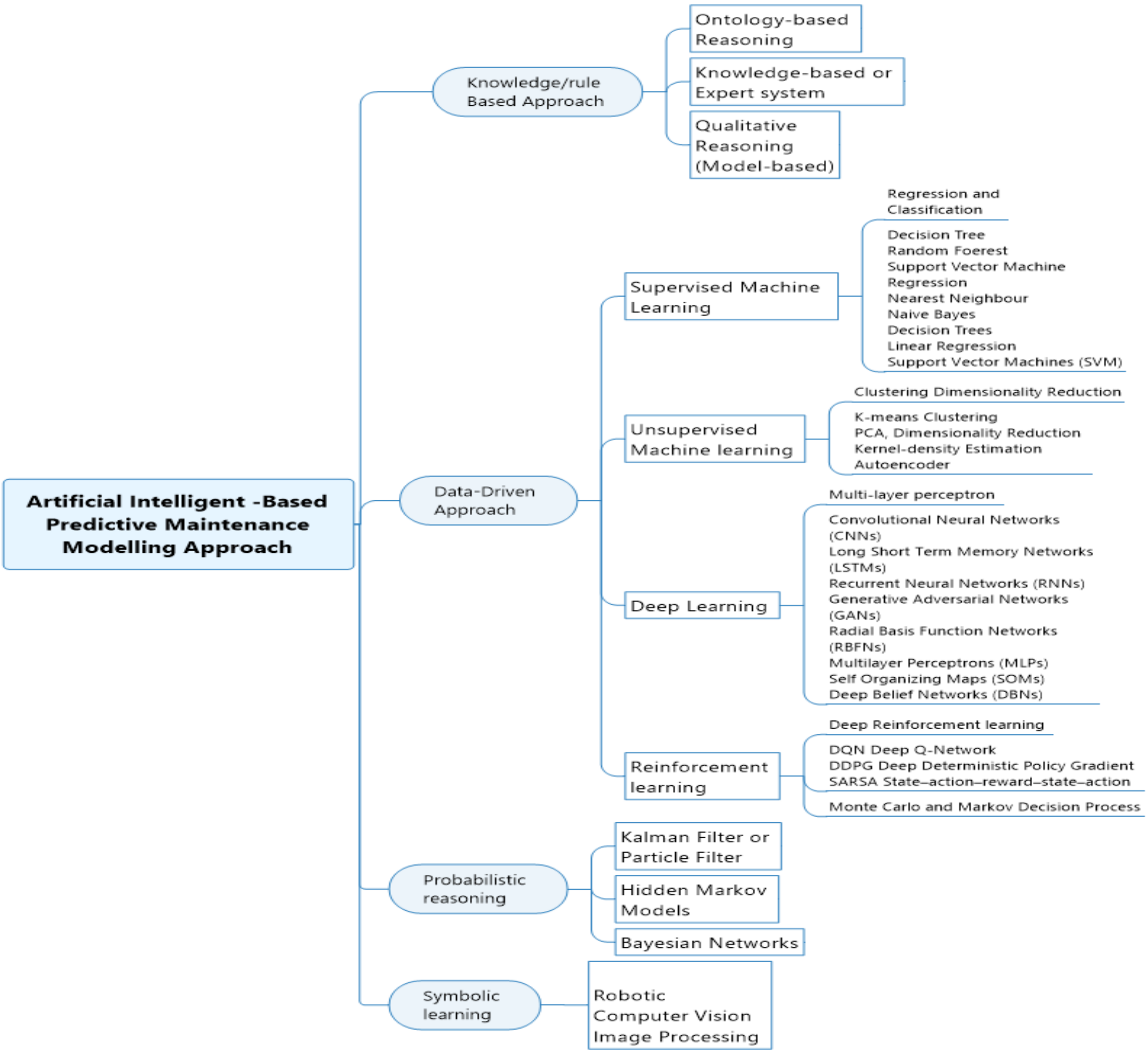

Artificial Intelligent -Based Predictive Maintenance Modelling Approach
Knowledge/rule Based Approach
Ontology-based Reasoning
Knowledge-based or Expert system
Qualitative Reasoning (Model-based)
Data-Driven Approach
Supervised Machine Learning
Regression and Classification
Decision Tree
Random Foerest
Support Vector Machine
Regression
Nearest Neighbour
Naive Bayes
Decision Trees
Linear Regression
Support Vector Machines (SVM)
Unsupervised Machine learning
Clustering Dimensionality Reduction
K-means Clustering
PCA, Dimensionality Reduction
Kernel-density Estimation
Autoencoder
Deep Learning
Multi-layer perceptron
Convolutional Neural Networks (CNNs)
Long Short Term Memory Networks (LSTMs)
Recurrent Neural Networks (RNNs)
Generative Adversarial Networks (GANs)
Radial Basis Function Networks (RBFNs)
Multilayer Perceptrons (MLPs)
Self Organizing Maps (SOMs)
Deep Belief Networks (DBNs)
Reinforcement learning
Deep Reinforcement learning
DQN Deep Q-Network
DDPG Deep Deterministic Policy Gradient
SARSA State–action–reward–state–action
Monte Carlo and Markov Decision Process
Probabilistic reasoning
Kalman Filter or Particle Filter
Hidden Markov Models
Bayesian Networks
Symbolic learning
Robotic
Computer Vision
Image Processing

Figure 3- 8 AI-Based Approach for Predictive Maintenance

Humans can see and process images with their eyes. Similarly, AI enables machines to recognise objects in their environment and generate a representation of that world.

The robotics industry gave the same opportunity for machines to become familiar with their surroundings and move around freely. Humans can group images together based on patterns, and AI can do the same using pattern recognition. The recent advances have provided machines with more capabilities to perform complex pattern recognition than humans [114] because more data with varying degrees of dimensions can be fed into machines to produce better results than humans. This field of AI is called data analytics and machine learning.

3.2.2 Application of AI and Big Data Analytics in the Aerospace Industry

In modern vehicles such as aircraft with installed IoT devices such as embedded sensors gave the capability to record more data covering many parameters (such as temperature, humidity, pressure, speed, altitude, stability, and other details during flight operation). The obtained data can be utilised to examine the health of various subsystems and components by providing failure records that can be used for aircraft diagnostics and prognostics as well as other ground-based maintenance tasks.

The aerospace industry can better understand the challenges of handling big data and its opportunities than other domains. According to Oliver Wyman's Survey of 2019 [104], it is shown that a single modern Jet engine can generate up to 10TB worth of data in 1:30 minutes of flight time, with many hundreds to- a thousand flights per day, data volume could reach up to many Petabytes. There could be challenges in transforming this type of big data for actionable insight, but the advantages are worthwhile. For instance, by deploying the right data analytics, airlines, OEMs, and MROs can maximise their operational efficiency, improve planning and strategically align decision making. Data-driven predictive maintenance models can also be deployed to mitigate unscheduled maintenance, which will help airlines maximise revenue by keeping their fleet up and running. The aviation industry's technological development is changing the way data is collected, stored, maintained, processed, and analysed for more informed insight. The nature of the data generated comes with newer analytical challenges that require newer approaches to handle them [115].

The aerospace industry's technology advancements have produced a highly competitive market, forcing airlines and other businesses to seek out new ways to improve service delivery [116]. The key is to continuously improve product quality and operational efficiency to remain viable and competitive [117]. As a result, every organization's goal is to improve vehicle availability and system reliability, which necessitates implementing a more cohesive and thorough maintenance programme at a greater expense. As a result, every company strives to reduce overall costs while maintaining efficient operations. They use artificial intelligence (AI) and other associated technologies to tap into the massive quantity of data accessible to understand their fleet's behaviour better. [118].

Big-data analytics benefits can be seen throughout the entire life cycle of aircraft development, from the early design stages to production to in-flight operations to maintenance support [119]. For example, data analytics is critical in designing and testing automobile engines and parts. Any risks that could have an impact on the vehicle or part quality can be found early by analysing data from the production cycle during testing. These investigations' findings can also be used to improve component designs in the future. Aircraft operators and maintenance engineers can utilise the performance and health data to find flaws and predict future occurrences. Airlines may also forecast when certain vehicle parts are likely to fail and do preventative maintenance. Onboard sensors create data that analytics systems can employ to collect performance and operational data while monitoring the health of aircraft components in the cockpit in real-time. Based on the aircraft's performance data, data analytics can also be utilised to optimise in-flight operations by estimating an arrival time, fuel consumption, and aircraft mass, among other things. These parameters can be utilised to optimise the flight route for the best possible fuel efficiency.

Many new data analytics technologies have continued to emerge in the aerospace industry. The Network Edge Analytics (NEA) capabilities are one of these solutions. NEA is a data analysis model in which incoming data streams are analysed in a non-central place such as a switchboard or connected notes before being moved to the core location for analyses [117]. The analytics solutions that can be implemented both at the network edge and network core are more advantageous. More reviews about NEA can be found in Bakshi et al. [115] and Satyanarayanan et al. [120]. A critical review of the techniques, tools, infrastructure and general application of data analytics for health monitoring, predictive analysis, and optimisation of aircraft performance can be found in Weerasinghe et al.[121]. Their study further shows the significant capability to address contemporary challenges in applying data analytics in aircraft. Data analytics framework for improving the quality, performance, and health monitoring of aircraft auxiliary power units (APU) is proposed by Xu et al.[122]. Likewise, Yang et al. [123] propose a big data platform for civil aircraft to facilitate civil aviation companies' operations. The architecture of the platform is based on the standards of cloud computing. It provides techniques to support decision-making, such as maintenance scheduling, prognostics alerts, diagnostics, fuel-saving, and airline schedules. Sciancalepore et al. [124] developed an IoT-based measurement system for monitoring aerial vehicles. It has the capability of coordinating a collection of large datasets from embedded devices installed on the aircraft.

In conclusion, this section provides a brief overview of the use of data analytics in the aerospace industry. Data is becoming more accessible as a result of technological advancements, but it also brings with it new analytical hurdles. These problems necessitate innovative solutions in order to extract knowledge from them and make better decisions. Many data analytics methodologies and solutions established for other industries, such as healthcare, banking, online applications, and so on, were also proven to be incompatible with the difficulties faced by the aerospace industry. More study on edge analytics for aviation use is also required. There is also no unified data analytics solution for network edge and core analytics in the free literature. This is because, as a competitive advantage strategy, businesses want to process their data internally.

3.3 Predictive Maintenance in Aerospace

Over time, aircraft maintenance procedures have evolved, but the purpose has remained the same: to protect the aircraft. Mainly, maintenance strategies can be categorised into three, failure-based (Reactive), time-based (preventive), and condition-based (predictive) maintenance.

Failure-based maintenance, also known as reactive maintenance, is carried out after a failure, as shown in Figure 3-9. Reactive maintenance is only appropriate for non-critical components because it increases unscheduled downtime and makes maintenance difficult to plan. Doing business with this form of maintenance is quite expensive, especially in a complicated and safety-critical system like an aircraft. Preventive maintenance was initially developed to overcome reactive maintenance challenges by periodically scheduling assessment and replacement of components [125]. In Preventive maintenance, maintenance activities can be scheduled, such that human resources requirements can be planned, spare parts can be ordered as needed. This can reduce events of unpredictable failure because the component can be replaced before failure [126]. Preventive maintenance can be optimised to reduce unnecessary failure, which in turn reduces downtime and associated cost. One of the significant challenges with preventive maintenance is knowing when to do maintenance since failure can occur even before the next scheduled repair. In planning for preventive maintenance, one must be careful, especially in a safety-critical system. Also, because preventive maintenance is time-based, components can be prematurely replaced. Resources can be underutilised if components are prematurely replaced, which adds to the cost [116,127]. Therefore, preventive maintenance is unable to completely remove unscheduled maintenance scenarios, which incurs additional maintenance cost.

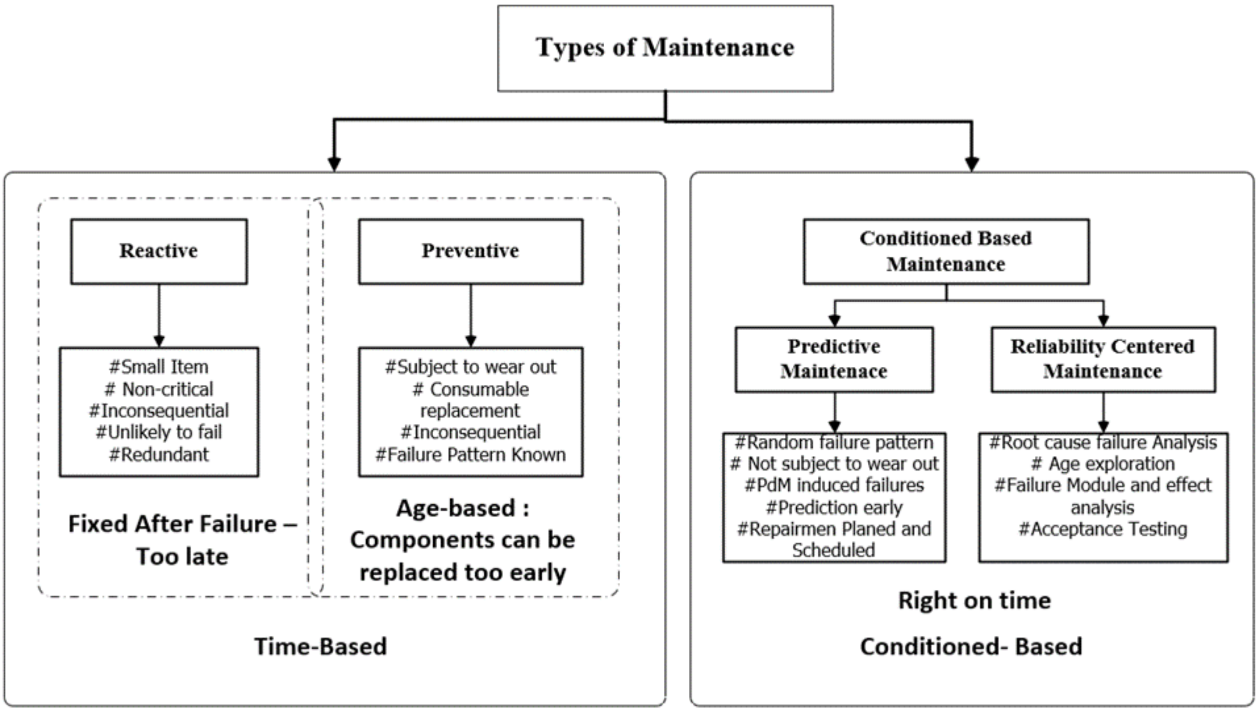


Figure 3- 9 Types of Maintenance

However, if machine failure can be predicted, maintenance can be scheduled right before it occurs.

Predictive maintenance tries to mitigate the drawbacks of preventive maintenance. The goal is to prevent unexpected failures by continually monitoring the health condition of aircraft components. This will enable estimation of time to failure, diagnose problems in complex vehicles, and help identify the parts that need to be fixed, minimising downtime and maximising vehicle life. Predictive maintenance requires the development of robust algorithms that are capable of predicting in advance a time when a component will fail and when maintenance will be required.

Many new study ideas have been offered in the literature, all of which focus on employing data analytics to improve aircraft maintenance. A systematic literature review about data analytics applications and related technologies in maintenance is shown in Buam et al. [128]. Their study aims to generally provide a literature-based evaluation of the application of big data analytics for maintenance. Moreover, a data analytics model for managing aircraft routing and maintenance staffing with price competition using the Stackelberg-Nash game algorithm developed by Eltoukhy et al.[129]. The authors designed the model to capture the interdependence between aircraft routing of airlines and maintenance providers' maintenance staffing to reduce cost savings for both airlines and maintenance providers. Likewise, Hiruta et al. [130] design a data analytics process for condition-based maintenance. Their work aims to bridge the gap between the data scientist and maintenance engineers in developing predictive maintenance models. They developed an engineering tool that specifies a workflow stating roles for data scientists and maintenance engineers in the modelling process. Puttini et al. [131] discuss how big data analytics can be incorporated into the Integrated Vehicle health management (IVHM) platform. They also present business benefits derived for such incorporation, which include but are not limited to optimisation in vehicle design, operation, and maintenance. Also, Nayak et al.[132] show how big avionic data (sensor data and LRU fault data) can be used to provide on-board and off-board prognostics using OSA-CBM and Hadoop framework. Similarly, a study conducted by Dubrawsk et al. describes how big data analytics techniques are used for public health surveillance to support aerospace fleet management [30] effectively.

Their study focuses on techniques for early warning of systematic failures of aerospace components. [133] proposed a platform for integrating big data technology into the process of civil aircraft health management.

3.3.1 The Role of IVHM in the Aerospace Industry

NASA first conceptualised Integrated Vehicle Health Management (IVHM) in 1992 in an article titled "research and technology goals and objectives for IVHM" [134]. It describes a set of unified systems to assess the current or future performance that enables effective and efficient health assessment of the target vehicle before, during, and after the operation. It accounts for collecting data relevant to an asset's present and future performance and transforming it into information to support operational decisions. This includes the ability to perform timely status determination, diagnostics, and prognostics [135–137]. IVHM technologies provide many advantages to vehicle management, such as reducing maintenance cost, provision of precise scheduled maintenance, real-time prognostics and diagnostics services, timely arrangement for spare parts, and provision of more realistic condition-based maintenance. Integrating AI in the IVHM system can increase the effectiveness of its applicability.

Internet of Things (IoT) has evolved tremendously in all spheres of our lives, ranging from social to industrial applications. The concept of IOT describes intelligent networking of physical smart devices (such as sensors, actuators, and switches) using the internet network, which enables them to collect and share data. IoT is already playing a significant role in aircraft maintenance and safety. Towards smart-maintenance, AI-Based IVHM algorithms can interpret and organise data from sensors and send the data in a report which can easily be comprehended[138]. This algorithm also identifies and reports on potential failures in real-time and arranges proper timelines for repairs. As seen in Figure 3-10, IoT enabled cyber-physical systems to generate data, and the data goes through a pre-processing phase to be transformed for machine learning. The performance of the predictive model is highly dependant on the pre-processing methods. After the data is transformed, AI-based reasoning algorithms are trained to produce a model that provides intelligent maintenance decisions.

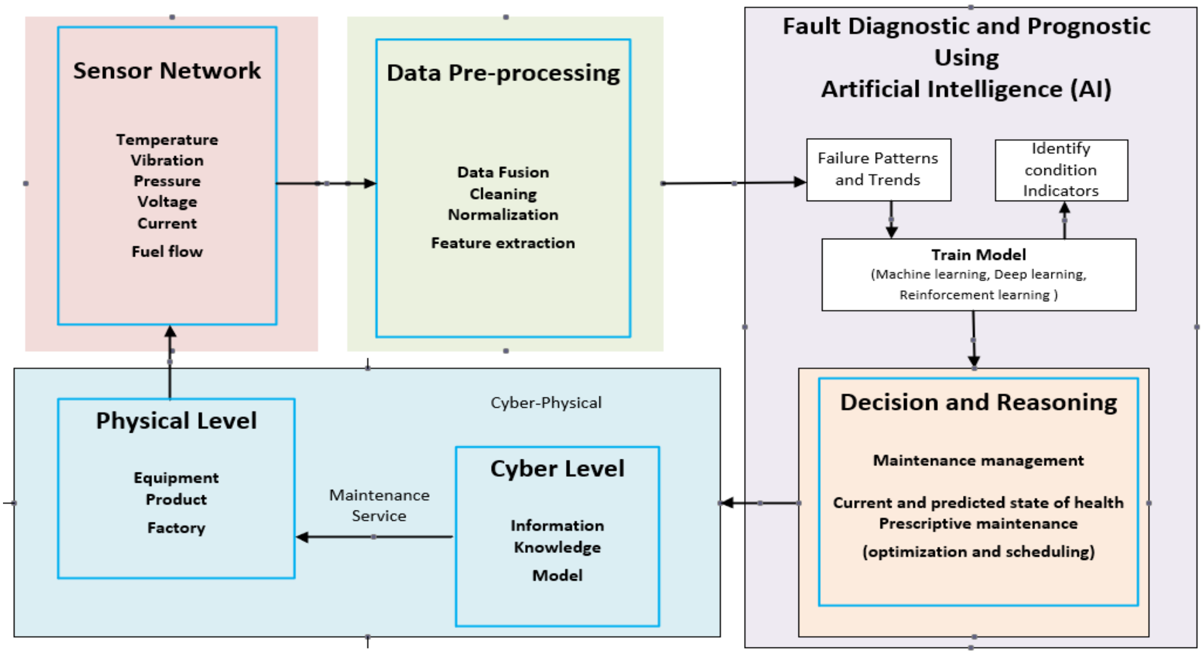


Figure 3- 10 Example of a Cyber-Physical System IVHM Framework

Many tools and software have been developed and successfully deployed using IVHM architecture. For example, the Livingstone -open-source model-based diagnostic reasoning tools developed by NASA Ames Research Centre (NASA-ARC) has been successfully implemented on X-34 and X-37 [139][140]. Integrated System Health Management (ISHM) and Beacon-based Exception Analysis for Multi-mission (BEAM), reasoning and diagnostic tools developed by NASA -ARC and the Jet Propulsion Laboratory (JPL), were successfully applied in the X-33 project [141]. Some aircraft condition monitoring system that has been developed and successfully implemented are as follows. The health and usage monitoring system (HUMS), the aircraft condition monitoring system (ACMS) [142], the engine monitoring system (EMS), IVHM and Engine Health Monitoring EHMS in the aircraft and engine systems [143], the integrated diagnostics and prognostics system (IDPs) [144,145], integrated condition assessment system (ICAS) [146]. Crew Information Service/Maintenance System (CIS/MS) was recently implemented in Boeing B787 aircraft. The CIS/MS is responsible for applications such as central maintenance systems and electronic flight bags. IVHM was named the Prognostic Health Management (PHM) and the Autonomic Logistics in the United States joint strike fighter JSF-35 [147]. In implementing the PHM - In-flight and connected with the joint distributed information system (JDIS) in the ground forms a complete IVHM system [148]. JDIS implementation was recognised as the highest level of U.S. military condition-based maintenance (CBM) technology [149]. Its operation and safety system is designed on-board, and the control of maintenance scheduling and fleet management is deployed on the ground station [148]. The logical reasoning system structure is adopted, and the inference engine is designed at the member, regional, and aircraft levels [149].

IVHM system's implementation has shown significant benefits to aircraft maintenance. However, some areas need further research. One of the challenges is the inter-system data compatibility resulting from inter-system connectivity in the IVHM ecosystem due to different manufacturers' architectural designs, hence having a different data pre-processing method. Therefore, integrating the data for analysis becomes challenging [150].

2.3.2 Digital Prescriptive Maintenance for Complex Systems

Before the advent of IoT, vehicle maintenance was mainly based on prearranged time-based schedules linked to the vehicle's age, the number of schedule cycles, or usage. It was not linked to the vehicle's real-time condition of the vehicle. Time-based maintenance is susceptible to unnecessary onsite vehicle inspections or visits to the service centre. Potential failures can go unnoticed between the schedules, and there are small or no useful insights for the OEM's and MRO's.

The concept of prescriptive maintenance in a complex system is gaining more research attention, especially in the aviation industry [151]. Prescriptive maintenance is advanced predictive maintenance. It leverages preventive, descriptive, and predictive maintenance approaches and capabilities to optimise system performance completely. Prescriptive maintenance taps the power of IoT, big data analysis, machine learning and dynamic case management to help vehicles become proactive participants in their maintenance [152]. This type of maintenance promises cost saving over time-based preventive maintenance because maintenance is carried out only when warranted. Figure 3-11 shows the functional flow for digital prescriptive maintenance for aircraft. As we can see, aircraft data from different sources recorded over time can be analysed in the data analysis platform by applying machine learning and other related technologies. The pre-processed data can then be used to train predictive models. Predictive models can predict when a component will fail so that maintenance can be planned. A prescriptive section can recommend the type of repairs needed. The smart Interaction can leverage IoT to adjust parameter based on the conditions of others automatically.

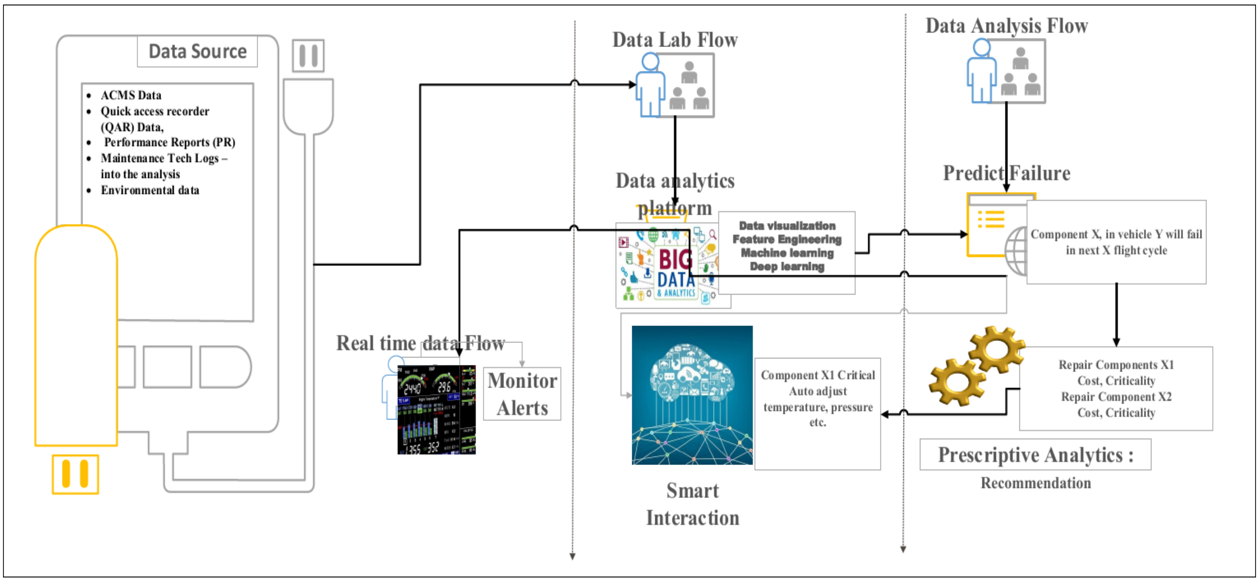


Figure 3- 11 Functional Flow for digital Prescriptive Maintenance for aircraft

Despite the advantage of predictive maintenance, most airlines still rely on preventive maintenance strategies. Preventative maintenance is a strategy where vehicle maintenance procedures are defined and scheduled to be performed periodically. However, when an unexpected failure occurs in-between the defined schedule period, the vehicle becomes unavailable until this problem is fixed. Predictive maintenance is developed to handle some of the drawbacks of preventive maintenance.

One of the design goals of predictive maintenance is to avoid unexpected failures by monitoring the vehicle condition and providing failure alerts well in advance. Predictive maintenance models are developed to forecast when likely the vehicle will fail, so that maintenance can be systematically scheduled to occur way in advance before the failure point. Predictive maintenance can be modelled in physics-based, knowledge-based, and data-driven-based [153]. Physics-based modelling can be defined as a simplified mathematical description of a system or process to assist calculations and predictions [154]. The prediction is based on a mathematical equation inside the mode; therefore, it uses a limited amount of data compared to other methods. However, the physics-based model is challenging to create and implement, especially for complex systems, because it is sensitive to the system's design and material properties. Also, enough component information and a good knowledge of the failure mechanism is highly required to formulate the model.

The knowledge-based model, also known as the expert system, uses defined rules or fuzzy logic to solve complex problems. The rules are set based on the knowledge of a domain expert. Converting domain knowledge to a set of rules is challenging, requiring another prognostics technique. Also, the set of rules needs to be updated anytime there is any system update. This process can be cumbersome and sometimes impractical, especially in a complex system with many components and processes.

The data-driven approach involves training machine learning algorithms using large historical datasets to automatically learn a system behaviour model. A data-driven approach is easy to implement, flexible, adaptable with a low cost of implementation. However, large historical data representing failure is needed, and getting such data is always challenging. However, the advancement in technology data is increasingly available, making it more appealing to use a data-driven approach for developing predictive maintenance models in complex systems. The hybrid of the two or three approaches to predictive maintenance is also possible [155].

In conclusion, both predictive maintenance modelling approaches have inadequacies in real-world scenarios, and there is no universally accepted predictive modelling approach. Adapting modelling is dependent on the scenario at hand. Data-driven and machine learning approaches are desirable in developing predictive models for complex systems such as aircraft. However, it requires a large dataset to provide desirable results.

3.3.3 Application of Machine Learning for Predictive Modelling

Data-driven predictive maintenance modelling depends majorly on machine learning techniques to build models. The use of machine learning for developing predictive models has significantly increased in recent times. The growth is due to technological advancement, which provides more computational power, processing speed, and improved data storage. The emerging AI technologies have also made processing larger, unstructured, and more complicated datasets easier and faster, producing models with improved performance. Machine learning is divided into the following major categories; supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Supervised learning is a type of machine learning which uses labelled data to train a learning algorithm. In other words, the input and input are known, and the algorithm learns by comparing the actual with correct inputs and finds errors. Supervised learning is mostly used for classification or regression problems. Unsupervised learning is a type of machine learning that draws an inference from an input dataset without label responses. Unsupervised learning is mostly used to find hidden patterns in a dataset—for example, the k-means algorithm partitions data into k distinct clusters based on the centroid's distance.

As seen in Figure 3-12, reinforcement learning is an area of machine learning involving an agent taking action in an environment, and a reward is returned for every action taken [156].

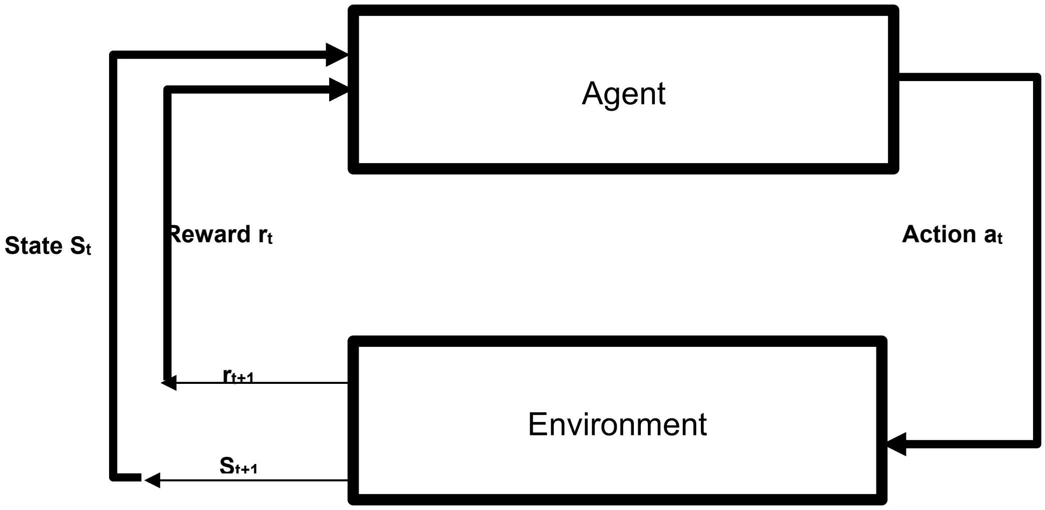


Figure 3- 12 Parameter Interaction in Reinforcement Learning

The 'environment' is the world through which the agent moves, 'State' is a concrete situation in which the agent finds itself, the reward is the feedback that measures the success and failure of an agent's action. Choosing the suitable type of machine learning approach to use is highly dependant on the challenge at hand.

Conditioned-based Maintenance (CBM): In CBM, sensors are installed to monitor a vehicle's health performance. Thresholds can be defined and subsequently be adjusted manually based on human-defined rules so that when sensor data violet thresholds rule, an alert is triggered to signal potential fault. This type of maintenance approach can only be feasible in monitoring a small number of embedded sensors or components. The component-based approach is limited to a complex system with many parts. Managing each component will be cumbersome and, in most cases, impractical. On the other hand, the machine-learning approach does not require rules or manual threshold settings. A large amount of data can be fed into a machine-learning algorithm and trained automatically to cluster similar components, identify patterns and correlations, and detect abnormalities. The algorithm can identify components degradation using related failures record and predict maintenance needs based on behavioural analysis.

Machine learning has been applied widely in many domains to solve complex analytical problems. Recently, substantial research and development in the aerospace industry have focused on optimising asset maintenance through AI or machine learning methods. For example, Wim et al.[157] developed a data-driven predictive model for aircraft component failure prediction. Also, predictive line maintenance optimisation of redundant aeronautical systems subjected to multiple wear conditions is developed by Wlamir et al.[158].

Finally, machine learning has proven to be beneficial in developing predictive maintenance models. However, to train the model for more accurate prediction, a vast amount of data is required. As a result of the recent advancements in artificial intelligence, more data is becoming available, allowing for the development of more efficient models. Despite these advantages, several discovered issues require further investigation, such as the data imbalance problem and high dimension space.

3.4 Research Gaps

This thesis considers the following research gaps.

Research Gap 1- Lack of algorithm to extract patterns log-based ACMS data for predictive maintenance model.

The large log-based dataset is a fruitful source of information for equipment diagnostics and prognostics; however, elaborate data pre-processing is necessary to harness such valuable pieces of information. Analysing the large system log to develop predictive models is the main challenge of data-driven predictive maintenance [4]. The raw ACMS data contains heterogeneous characteristics, including numeric time series, symbolic sequence, categorical variables, and unstructured text, requiring an intensive and meticulous pre-processing approach. The challenge of predicting rare failure using a large log-based time-series dataset is that the data distribution has irregular patterns and trends, which affects the learning of temporal features. Some data complexities coexist with imbalance problems, which can negatively impact the performance of data-driven models, such as noise, outliers, class overlapping, small class disjuncts, small sample size, and extreme minority class.

Research Gap 2- Most studies for rare failure prediction problems for the vehicle predictive maintenance model usually validate their work on a slightly imbalanced dataset while extreme failure cases are ignored: Frequent events create an imbalanced dataset. Meaning having significantly fewer samples in one class (say positively labelled) than other classes with large samples (say negatively labelled). The positively labelled data can be around 5–10% of the total dataset in a slightly rare event problem. In an extremely rare event problem, the positively labelled data can be 1% or less of the whole dataset. Training machine learning algorithms with an extremely imbalanced dataset with irregular patterns and trends can cause biases in the classification process, resulting in model performance degradation. Hence, producing a high false-positive rate and imprecise prognostics. Most of the open literature research validates its work on a slightly imbalanced ratio of the total dataset. However, in some real-world problems, such as big data, it can contain an imbalance ratio of up to 1% or less; in such cases, the existing solution becomes limited.

Research Gap 3-Extremely imbalanced dataset with class overlapping and Small class disjunct problem in ACMS dataset: - Diversity analysis is required to investigate the variation and bias rate and determine its effects in imbalanced ensemble learning models. There is no clear indicator of how classifiers should be constructed and connected in a large data domain. Also, the open literature lacks a study that investigates the impact of the combination of class Imbalance, small disjuncts and skewed class distribution on classifier performance, especially using the

ACMS dataset. Also, existing methods for handling slightly imbalanced datasets are understandable for certain types of datasets (such as image classification). However, because there may be no precise temporal contexts and observable in text-sequence learning, it is questionable whether training severely imbalanced, heterogeneous time-series data using the existing approach may increase model performance.

3.4.2 Summary of other potential Gaps in the imbalance learning

The following research gaps are future directions that the thesis does not cover.

1. Analysing the structure of classes present in the dataset: In the analysis of the class structures, it was discovered that there exist some data complexities that coexist with imbalance problems which can negatively impact the performance of data-driven models, such as noise, outliers, class overlapping, small dis-jaunts, small sample size, extreme minority class.

Suggested solution: Develop a new machine learning algorithm to incorporate data structure checking, especially the neighbourhood of minority class data points. The learning algorithm should stop bias towards the majority class and be able to handle small dis-jaunts inherently. A previous study has shown how to investigate nosily and outlier roles in minority classes Napierala et al. [67]. The study suggested dropping data points that are considered noise or outliers, but losing more points may not be a good idea considering the small data size. Other preliminary work shows a high potential for accurate filtering of noise and outliers in a cluster, which can then be safely dropped [159–161]. Filtering noise in rare event prediction or learning from the imbalanced dataset can be investigated for algorithm optimisation.

2. Multi-class imbalanced classification problem: There is a situation where the classes present in a dataset are more than two, known as multi-class. A multi-class imbalanced classification problem exists in many forms. It could be one majority class to multi-minority or one majority to some normal minority and some extreme minority. Another form can be a multi majority to one or more minorities. Many studies in the open literature have been done in two-class problems, with less attention to multi-class problems. Previous studies have tried to simplify the problem by decomposing the multi-class into sub binary classes and then combining the result after that (one vs one or one vs all techniques)[69]. However, there is a need to have a unified solution to handle all the varied forms of multi-class imbalanced problems.

Suggested Solution: Develop a different solution, either one-vs-one or one-vs-all techniques, like a high number of base classifiers or introducing additional artificial imbalance [70]. Develop a method to consider varying relationships between classes in multi-class imbalance learning. Considering the algorithm level approach, design a classifier or modify the existing classifier to handle multi-class imbalance problems and be insensitive to skewness. An algorithm level approach is seen in Cieslak et al. [163]; the authors show how the decision tree algorithm can be modified to accommodate the multi-class imbalance problem. Similarly, Yu et al.[79] transform neural networks and proposed an ensemble solution for multi-class by decomposing it into classes. The approach suffers a drawback as the same approach to all the decomposed classes without considering when there are complexities in the distribution (overlapping, multi-majority, multi-minority, class noise), which can result in inconsistency in the pairwise relationship

4. Imbalanced Regression problem (predicting continuous target variable): -The imbalance perspective of regression is yet to be explored exhaustively, which concerns the prediction of rare and extreme values of a continuous target variable. Methods to identify the difference between noise and outlier in imbalanced regression problems is needed.

Suggested Solution: Investigate the solution of imbalance classification problems for possible application to imbalance regression problems. Some previous work is Torgo et al. work on evaluation metrics for continued value [166] and SMOTE for Regression [167].

5. High Dimensionality: The continuous development of IoT enabled cyber-physical systems and made more data available. The dataset comes with newer analytics challenges such as high and multi-dimensional. For example, sensors embedded in an aircraft to monitor many components and other systems configuring could give the resulting dataset a high dimension. A large number of variables or features in an observation can affect the learning algorithm's performance. It becomes more challenging in the case where the dataset is extremely imbalanced.

Suggested Solution: Create a method for reducing features in large data sets with many dimensions that is both efficient and effective. One example is combining feature reduction techniques like PCA, LDA, or Autoencoder with resampling or cost-sensitive algorithms. Consider employing a deep learning strategy.

6. Imbalance learning for data streams (concept drift): In online learning, imbalanced data that is in batches or online poses new challenges because of its dynamic nature (changing in imbalance ratio, the relationship between classes, general underlying distribution); this is known as concept drift. An adaptive method approach is needed to deal with skew data coming in online in real-time. Challenge of drifting, which affects the class distribution Getting the character and structure of minority class in streaming data, is not easy as static.

Suggested solution: Investigate the effects of multi-label or multi-instance and ensemble learning in the concept drift domain. Consider approaches dedicated to storing general solutions for streams instead of reacting to each reappearance of class imbalance anew. The stored classifiers can be used when similar distribution reappears. Investigate the nature of the imbalance problem in the data stream and shifting drift consider multi-class [168][169].

3.5 Reference

1. Branco P., Torgo L., Ribeiro R. A Survey of Predictive Modelling under Imbalanced Distributions. 2015; : 1–48. Available at: DOI:10.1145/2907070

2. Eickmeyer J., Li P., Givehchi O., Pethig F., Niggemann O. Data Driven Modeling for System-Level Condition Monitoring on Wind Power Plants. Int. Work. Princ. Diagnosis. 2015; 1507: 43–50.

3. Ossai C. Integrated Big Data Analytics Technique for Real-Time Prognostics, Fault Detection and Identification for Complex Systems. Infrastructures. 2017; 2(4): 20. Available at: DOI:10.3390/infrastructures2040020

4. Wagner C., Saalmann P., Hellingrath B. Machine Condition Monitoring and Fault Diagnostics with Imbalanced Data Sets based on the KDD Process. IFAC-PapersOnLine. Elsevier B.V.; 2016; 49(30): 296–301. Available at: DOI:10.1016/j.ifacol.2016.11.151

5. Branco P., Torgo L., Ribeiro RP. A Survey of Predictive Modeling on Imbalanced Domains. ACM Computing Surveys. 2016; 49(2): 1–50. Available at: DOI:10.1145/2907070

6. Nghiem LT., Thu TT., Nghiem TT. MASI: Moving to adaptive samples in imbalanced credit card dataset for classification. 2018 IEEE International Conference on Innovative Research and Development, ICIRD 2018. 2018. pp. 1–5. Available at: DOI:10.1109/ICIRD.2018.8376315

7. Sajana T., Narasingarao MR. A comparative study on imbalanced malaria disease diagnosis using machine learning techniques. Journal of Advanced Research in Dynamical and Control Systems. 2018; 10: 552–561. Available at: DOI:https://www.jardcs.org/backissues/abstract.php?archiveid=2962&action=fulltext&uri=/backissues/abstract.php?archiveid=2962

8. Jiao Z., Jia G., Cai Y. A new approach to oil spill detection that combines deep learning with unmanned aerial vehicles. Computers and Industrial Engineering. Elsevier; 2018; (September): 1–12. Available at: DOI:10.1016/j.cie.2018.11.008

9. Yusof R., Kasmiran KA., Mustapha A., Mustapha N., Zin NAM. Techniques for handling imbalanced datasets when producing classifier models. J. Theor. Appl. Inf. Technol. 2017; 95(7): 1425–1440.

10. Douzas G., Bacao F. Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Systems with Applications. Elsevier Ltd; 2018; 91: 464–471. Available at: DOI:10.1016/j.eswa.2017.09.030

11. He H. Imbalanced Learning. Self-Adaptive Systems for Machine Intelligence. New Jersey: John Wiley & Sons, Inc.,Hoboken, New Jersey.; 2011. 44–107 p. Available at: DOI:10.1002/9781118025604.ch3

12. López V., Fernández A., García S., Palade V., Herrera F. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences. Elsevier Inc.; 2013; 250: 113–141. Available at: DOI:10.1016/j.ins.2013.07.007

13. Haixiang G., Yijing L., Shang J., Mingyun G., Yuanyue H., Bing G. Learning from class-imbalanced data: Review of methods and applications. Expert Syst. Appl. Elsevier Ltd; 2017; 73: 220–239. Available at: DOI:10.1016/j.eswa.2016.12.035

14. Elrahman SMA., Abraham A. A Review of Class Imbalance Problem. Netw. Innov. Comput. 2013; 1: 332–340. Available at: DOI:www.mirlabs.net/jnic/index.html

15. Haixiang G., Yijing L., Shang J., Mingyun G., Yuanyue H., Bing G. Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications. 2017; 73: 220–239. Available at: DOI:10.1016/j.eswa.2016.12.035

16. Johnson JM., Khoshgoftaar TM. Survey on deep learning with class imbalance. Journal of Big Data. Springer International Publishing; 2019; 6(1). Available at: DOI:10.1186/s40537-019-0192-5

17. Wu Z., Lin W., Ji Y. An Integrated Ensemble Learning Model for Imbalanced Fault Diagnostics and Prognostics. IEEE Access. 2018; 6: 8394–8402. Available at: DOI:10.1109/ACCESS.2018.2807121

18. Krawczyk B., Woźniak M., Schaefer G. Cost-sensitive decision tree ensembles for effective imbalanced classification. Applied Soft Computing Journal. Elsevier; 1 January 2014; 14(PART C): 554–562. Available at: DOI:10.1016/j.asoc.2013.08.014 (Accessed: 6 February 2019)

19. Project S., Belkhayat K., Omar A. XGBoost and LGBM for Porto Seguro ' s Kaggle challenge : A comparison. 2018;

20. He H., Garcia EA. Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering. 2009; 21(9): 1263–1284. Available at: DOI:10.1109/TKDE.2008.239

21. Chawla N V., Bowyer KW., Hall LO., Kegelmeyer WP. SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research. 2002; 16: 321–357. Available at: DOI:10.1613/jair.953

22. Fernández Alberto, Garcia Salvador, Galar Mikel, Prati Ronaldo, Krawczyk Bartosz HF. Learning From Imbalanced Data Sets. 2018. Available at: DOI:https://link.springer.com/content/pdf/10.1007%2F978-3-319-98074-4.pdf (Accessed: 6 May 2019)

23. Wilson DL. Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. IEEE Transactions on Systems, Man and Cybernetics. 1972; 2(3): 408–421. Available at: DOI:10.1109/TSMC.1972.4309137

24. Structures MB. kNN approach to imbalanced data distributions: a case study involving information extraction. Proceedings of workshop on learning from imbalanced datasets. 2003; 126(3).

25. Das B., Krishnan NC., Cook DJ. Handling imbalanced and overlapping classes in smart environments prompting dataset. Data Min. Serv. 2014; : 199–219. Available at: DOI:10.1007/978-3-642-45252-9

26. Batista GEAPA., Prati RC., Monard MC. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. SIGKDD Explor. Newsl. 2004; 6(1): 20–29. Available at: DOI:10.1145/1007730.1007735

27. Tomek I. Tomek Link: Two Modifications of CNN. IEEE Trans. Systems, Man and Cybernetics. 1976; : 769–772. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4309452

28. Prati RC., Batista GEAPA., Monard., C. M. Class imbalances versus class overlapping: an analysis of a learning system behavior. MICAI 2004 Adv. Artif. Intell. 2004; (0): 312. Available at: https://link.springer.com/chapter/10.1007/978-3-540-24694-7_32

29. Tang Y., Gao J. Improved classification for problem involving overlapping patterns. IEICE Transactions on Information and Systems. 2007; E90-D(11): 1787–1795. Available at: DOI:10.1093/ietisy/e90-d.11.1787

30. Trappenberg TP., Back AD. Classification scheme for applications with ambiguous data. Proceedings of the International Joint Conference on Neural Networks. 2000; 6: 296–301. Available at: DOI:10.1109/ijcnn.2000.859412

31. Xiong H., Wu J., Liu L. Classification with ClassOverlapping: A Systematic Study. 2010; : 491–497. Available at: DOI:10.2991/icebi.2010.43

32. Zhang C., Tan KC., Ren R. Training cost-sensitive Deep Belief Networks on imbalance data problems. Proceedings of the International Joint Conference on Neural Networks. 2016. pp. 4362–4367. Available at: DOI:10.1109/IJCNN.2016.7727769

33. Park S., Lim J., Jeon Y., Choi JY. Influence-Balanced Loss for Imbalanced Visual Classification. 2021; : 735–744. Available at: http://arxiv.org/abs/2110.02444

34. Rofifah D. Pattern Classification Using Ensemble Methods. Paper Knowledge . Toward a Media History of Documents. 2020. 12–26 p.

35. Ling CX., Yang Q., Wang J., Zhang S. Decision trees with minimal costs. Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004. 2004; (Icml): 544–551. Available at: DOI:10.1145/1015330.1015369

36. Bradford JP., Kunz C., Kohavi R., Brunk C., Brodley CE. Pruning decision trees with misclassification costs. Lecture Notes in Computer Science (including

subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 1998; 1398: 131–136. Available at: DOI:10.1007/bfb0026682

37. Turney PD. Cost-Sensitive Classification: Empirical Evaluation of a Hybrid Genetic Decision Tree Induction Algorithm. Journal of Artificial Intelligence Research. 1995; 2: 369–409. Available at: DOI:10.1613/jair.120

38. Zhang Z., Pfister T. Learning Fast Sample Re-weighting Without Reward Data. Iccv. 2021; Available at: DOI:https://arxiv.org/abs/2109.03216v1

39. Chang EY., Li B., Wu G., Goh K. Statistical Learning for Effective Visual Information Retrieval. IEEE International Conference on Image Processing. 2003; 3: 609–612. Available at: DOI:10.1109/icip.2003.1247318

40. Dogan T., Uysal AK. On Term Frequency Factor in Supervised Term Weighting Schemes for Text Classification. Arabian Journal for Science and Engineering. Springer Berlin Heidelberg; 2019; 44(11): 9545–9560. Available at: DOI:10.1007/s13369-019-03920-9

41. Cui Y., Jia M., Lin TY., Song Y., Belongie S. Class-balanced loss based on effective number of samples. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2019; 2019-June: 9260–9269. Available at: DOI:10.1109/CVPR.2019.00949

42. Wang YX., Ramanan D., Hebert M. Learning to model the tail. Advances in Neural Information Processing Systems. 2017; 2017-Decem(Nips): 7030–7040.

43. Robinson A., Cook RD., Weisberg S. Residuals and Influence in Regression. Journal of the Royal Statistical Society. Series A (General). 1984. 108 p. Available at: DOI:10.2307/2981746

44. Dong Q., Gong S., Zhu X. Class Rectification Hard Mining for Imbalanced Deep Learning. Proceedings of the IEEE International Conference on Computer Vision. 2017. pp. 1869–1878. Available at: DOI:10.1109/ICCV.2017.205

45. Lin TY., Goyal P., Girshick R., He K., Dollar P. Focal Loss for Dense Object Detection. Proceedings of the IEEE International Conference on Computer

Vision. 2017; 2017-Octob: 2999–3007. Available at: DOI:10.1109/ICCV.2017.324

46. Vanesa Sancho E. TEMA 6 LA CLASIFICACIÓN DE LOS SERES VIVOS Contenidos. 2011 International Conference on Computer Vision. 2011; : 89–96. Available at: http://iesdionisioaguado.org/joomla/Distancia/ccnn/tema6clasificacionseresvivos.pdf

47. Hospedales TM., Antoniou A., Micaelli P., Storkey AJ. Meta-Learning in Neural Networks: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2021; : 1–20. Available at: DOI:10.1109/TPAMI.2021.3079209

48. Thrun S., Pratt L. Learning to Learn: Introduction and Overview. Learning to Learn. 1998; : 3–17. Available at: DOI:10.1007/978-1-4615-5529-2_1

49. Wu J., Xiong W., Wang WY. Learning to learn and predict: A meta-learning approach for multi-label classification. EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference. 2020; : 4354–4364. Available at: DOI:10.18653/v1/d19-1444

50. Krawczyk B., Woźniak M., Schaefer G. Cost-sensitive decision tree ensembles for effective imbalanced classification. Applied Soft Computing Journal. 2014; 14(PART C): 554–562. Available at: DOI:10.1016/j.asoc.2013.08.014

51. Vanschoren J. Meta-Learning: A Survey. 2018; : 1–29. Available at: http://arxiv.org/abs/1810.03548

52. Gressling T. 84 Automated machine learning. Data Science in Chemistry. 2020. 409–411 p. Available at: DOI:10.1515/9783110629453-084

53. Shu J., Xie Q., Yi L., Zhao Q., Zhou S., Xu Z., et al. Meta-weight-net: Learning an explicit mapping for sample weighting. Advances in Neural Information Processing Systems. 2019; 32(NeurIPS): 1–12.

54. Olowookere TA., Adewale OS. A framework for detecting credit card fraud with

cost-sensitive meta-learning ensemble approach. Scientific African. Elsevier B.V.; 2020; 8: e00464. Available at: DOI:10.1016/j.sciaf.2020.e00464

55. Kulluk S., Özbakir L., Tapkan PZ., Baykasoğlu A. Cost-sensitive meta-learning classifiers: MEPAR-miner and DIFACONN-miner. Knowledge-Based Systems. 2016; 98: 148–161. Available at: DOI:10.1016/j.knosys.2016.01.025

56. Seliya N., Abdollah Zadeh A., Khoshgoftaar TM. A literature review on one-class classification and its potential applications in big data. Journal of Big Data. Springer International Publishing; 2021. Available at: DOI:10.1186/s40537-021-00514-x

57. Liu Z., Wei P., Jiang J., Cao W., Bian J., Chang Y. MESA: Boost ensemble imbalanced learning with MEta-SAmpler. Advances in Neural Information Processing Systems. 2020; 2020-Decem(I).

58. Ren J., Yu C., Sheng S., Ma X., Zhao H., Yi S., et al. Balanced meta-softmax for long-tailed visual recognition. Advances in Neural Information Processing Systems. 2020; 2020-Decem(NeurIPS): 1–24.

59. Schulz H. Patt. Religion und Konflikt. 2011. 185–206 p. Available at: DOI:10.13109/9783666604409.185

60. Stefanowski J. Challenges in Computational Statistics and Data Mining. 2016. Available at: DOI:10.1007/978-3-319-18781-5

61. Kotsiantis S., Kanellopoulos D., Pintelas P. Handling imbalanced datasets : A review. Science. 2006; 30(1): 25–36. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.9248&amp;rep=rep1&amp;type=pdf

62. Raskutti B., Kowalczyk A. Extreme re-balancing for SVMs. ACM SIGKDD Explorations Newsletter. 2004; 6(1): 60–69. Available at: DOI:10.1145/1007730.1007739

63. Yamaguchi A., Nishikawa T. One-Class Learning Time-Series Shapelets. Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018.

IEEE; 2019; : 2365–2372. Available at: DOI:10.1109/BigData.2018.8622409

64. Mauceri S., Sweeney J., McDermott J. Dissimilarity-based representations for one-class classification on time series. Pattern Recognition. Elsevier Ltd; 2020; 100: 107122. Available at: DOI:10.1016/j.patcog.2019.107122

65. Zhou ZH. Ensemble methods: Foundations and algorithms. Ensemble Methods: Foundations and Algorithms. 2012. 1–218 p. Available at: DOI:10.1201/b12207 (Accessed: 31 January 2019)

66. Sun J., Lang J., Fujita H., Li H. Imbalanced enterprise credit evaluation with DTE-SBD: Decision tree ensemble based on SMOTE and bagging with differentiated sampling rates. Information Sciences. 2018; 425: 76–91. Available at: DOI:10.1016/j.ins.2017.10.017

67. Feng W., Huang W., Ren J. Class imbalance ensemble learning based on the margin theory. Applied Sciences (Switzerland). 2018; 8(5). Available at: DOI:10.3390/app8050815

68. Lu W., Li Z., Chu J. Adaptive Ensemble Undersampling-Boost: A novel learning framework for imbalanced data. J. Syst. Softw. Elsevier Inc.; 2017; 132: 272–282. Available at: DOI:10.1016/j.jss.2017.07.006

69. Yuan X., Xie L., Abouelenien M. A regularized ensemble framework of deep learning for cancer detection from multi-class, imbalanced training data. Pattern Recognition. 2018; 77: 160–172. Available at: DOI:10.1016/j.patcog.2017.12.017

70. Krawczyk B. Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence. Springer Berlin Heidelberg; 2016; 5(4): 221–232. Available at: DOI:10.1007/s13748-016-0094-0

71. Galar M., Fernandez A., Barrenechea E., Bustince H., Herrera F. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews. 2012; 42(4): 463–484. Available at: DOI:10.1109/TSMCC.2011.2161285

72. Wozniak M. Hybrid Classifiers. Studies in Computational Intelligence. 2014. 1–233 p. Available at: http://www.scopus.com/inward/record.url?eid=2-s2.0-84885105209&partnerID=tZOtx3y1

73. Galar M., Fernández A., Barrenechea E., Herrera F. EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. Pattern Recognition. 2013; 46(12): 3460–3471. Available at: DOI:10.1016/j.patcog.2013.05.006

74. López V., Fernández A., Moreno-Torres JG., Herrera F. Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics. Expert Systems with Applications. 2012; 39(7): 6585–6608. Available at: DOI:10.1016/j.eswa.2011.12.043

75. Liu C., Ren Y., Liang M., Gu Z., Wang J., Pan L., et al. Detecting overlapping data in system logs based on ensemble learning method. Wireless Communications and Mobile Computing. 2020; 2020(ii). Available at: DOI:10.1155/2020/8853971

76. Pouyanfar S., Tao Y., Mohan A., Tian H., Kaseb AS., Gauen K., et al. Dynamic Sampling in Convolutional Neural Networks for Imbalanced Data Classification. Proceedings - IEEE 1st Conference on Multimedia Information Processing and Retrieval, MIPR 2018. 2018. pp. 112–117. Available at: DOI:10.1109/MIPR.2018.00027

77. Lee H., Park M., Kim J. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. Proceedings - International Conference on Image Processing, ICIP. IEEE; 2016; 2016-Augus: 3713–3717. Available at: DOI:10.1109/ICIP.2016.7533053

78. Wang S., Liu W., Wu J., Cao L., Meng Q., Kennedy PJ. Training deep neural networks on imbalanced data sets. Proceedings of the International Joint Conference on Neural Networks. IEEE; 2016; 2016-Octob: 4368–4374. Available at: DOI:10.1109/IJCNN.2016.7727770

79. Buda M., Maki A., Mazurowski MA. A systematic study of the class imbalance

problem in convolutional neural networks. Neural Networks. Elsevier Ltd; 2018; 106: 249–259. Available at: DOI:10.1016/j.neunet.2018.07.011

80. Hensman P., Masko D. The Impact of Imbalanced Training Data for Convolutional Neural Networks. PhD. 2015; Available at: https://www.kth.se/social/files/588617ebf2765401cfcc478c/PHensmanDMasko_dkand15.pdf

81. Lee H., Park M., Kim J. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. Proceedings - International Conference on Image Processing, ICIP. IEEE; 2016; 2016-August: 3713–3717. Available at: DOI:10.1109/ICIP.2016.7533053

82. Pouyanfar S., Tao Y., Mohan A., Tian H., Kaseb AS., Gauen K., et al. Dynamic Sampling in Convolutional Neural Networks for Imbalanced Data Classification. Proceedings - IEEE 1st Conference on Multimedia Information Processing and Retrieval, MIPR 2018. IEEE; 2018; : 112–117. Available at: DOI:10.1109/MIPR.2018.00027

83. Buda M., Maki A., Mazurowski MA. A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks. 2018; 106: 249–259. Available at: DOI:10.1016/j.neunet.2018.07.011

84. Khan SH., Hayat M., Bennamoun M., Sohel FA., Togneri R. Cost-sensitive learning of deep feature representations from imbalanced data. IEEE Transactions on Neural Networks and Learning Systems. 2018; 29(8): 3573–3587. Available at: DOI:10.1109/TNNLS.2017.2732482

85. Wang H., Cui Z., Chen Y., Avidan M., Abdallah A Ben., Kronzer A. Predicting Hospital Readmission via Cost-Sensitive Deep Learning. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2018; 15(6). Available at: DOI:10.1109/TCBB.2018.2827029

86. Keisuke Nemoto , Ryuhei Hamaguchi , Tomoyuki Imaizumi SH. CLASSIFICATION OF RARE BUILDING CHANGE USING CNN WITH MULTI-CLASS FOCAL LOSS Keisuke Nemoto , Ryuhei Hamaguchi , Tomoyuki

Imaizumi , Shuhei Hikosaka Satellite Business Division , PASCO CORPORATION (Japan). 2018; : 4667–4670.

87. Keneshloo Y., Shi T., Ramakrishnan N., Reddy CK. Deep Reinforcement Learning for Sequence-to-Sequence Models. IEEE Transactions on Neural Networks and Learning Systems. 2019; : 1–21. Available at: DOI:10.1109/tnnls.2019.2929141

88. Van Hasselt H., Guez A., Silver D. Deep reinforcement learning with double Q-Learning. 30th AAAI Conference on Artificial Intelligence, AAAI 2016. 2016; : 2094–2100.

89. Lin E., Chen Q., Qi X. Deep reinforcement learning for imbalanced classification. Applied Intelligence. 2020; Available at: DOI:10.1007/s10489-020-01637-z

90. Arulkumaran K., Deisenroth MP., Brundage M., Bharath AA. Deep reinforcement learning: A brief survey. IEEE Signal Processing Magazine. 2017; 34(6): 26–38. Available at: DOI:10.1109/MSP.2017.2743240

91. Zhinin-Vera L., Chang O., Valencia-Ramos R., Velastegui R., Pilliza GE., Socasi FQ. Q-Credit Card Fraud Detector for Imbalanced Classification using Reinforcement Learning. ICAART 2020 - Proceedings of the 12th International Conference on Agents and Artificial Intelligence. 2020; 1(February): 279–286. Available at: DOI:10.5220/0009156102790286

92. Feng J., Huang M., Zhao L., Yang Y., Zhu X. Reinforcement learning for relation classification from noisy data. 32nd AAAI Conference on Artificial Intelligence, AAAI 2018. 2018; : 5779–5786.

93. Coralie M., Perrin G., Ramasso E., Rombaut M., Coralie M., Perrin G., et al. A deep reinforcement learning approach for early classification of time series To cite this version : HAL Id : hal-01825472 A deep reinforcement learning approach for early classification of time series. 2018 26th European Signal Processing Conference (EUSIPCO). EURASIP; 2018; : 2030–2034.

94. Hashemi, Lida Abdi S. An Ensemble Pruning Approach Based on Reinforcement Learning in Presence of Multi-class Imbalanced Data. Advances in Intelligent

Systems and Computing. 2014. 583–594 p. Available at: DOI:10.1007/978-81-322-1771-8_27

95. Zheng A., Casari A. Feature engineering for machine learning. O'Reilly Media. 2018. 218 p. Available at: https://perso.limsi.fr/annlor/enseignement/ensiie/Feature_Engineering_for_Machine_Learning.pdf%0Ahttps://www.amazon.com/Feature-Engineering-Machine-Learning-Principles/dp/1491953241

96. Devi S S., G P. Feature Engineering based Approach for Prediction of Movie Ratings. International Journal of Information Engineering and Electronic Business. 2019; 11(6): 24–31. Available at: DOI:10.5815/ijieeb.2019.06.04

97. Hameed SS., Petinrin OO., Hashi AO., Saeed F. Filter-wrapper combination and embedded feature selection for gene expression data. International Journal of Advances in Soft Computing and its Applications. 2018; 10(1): 90–105.

98. Kasongo SM., Sun Y. A deep learning method with filter based feature engineering for wireless intrusion detection system. IEEE Access. IEEE; 2019; 7: 38597–38607. Available at: DOI:10.1109/ACCESS.2019.2905633

99. Wang Y., Yao H., Zhao S. Auto-encoder based dimensionality reduction. Neurocomputing. Elsevier; 2016; 184: 232–242. Available at: DOI:10.1016/j.neucom.2015.08.104

100. Chen CY., Leu JS., Prakosa SW. Using autoencoder to facilitate information retention for data dimension reduction. IGBSG 2018 - 2018 International Conference on Intelligent Green Building and Smart Grid. IEEE; 2018; : 1–5. Available at: DOI:10.1109/IGBSG.2018.8393545

101. Smallman L., Artemiou A. A study on imbalance support vector machine algorithms for sufficient dimension reduction. Communications in Statistics - Theory and Methods. Taylor & Francis; 2017; 46(6): 2751–2763. Available at: DOI:10.1080/03610926.2015.1048889

102. Nanni L., Fantozzi C., Lazzarini N. Coupling different methods for overcoming the class imbalance problem. Neurocomputing. Elsevier; 2015; 158: 48–61.

Available at: DOI:10.1016/j.neucom.2015.01.068

103. Zhu R., Guo Y., Xue JH. Adjusting the imbalance ratio by the dimensionality of imbalanced data. Pattern Recognition Letters. Elsevier B.V.; 2020; 133: 217–223. Available at: DOI:10.1016/j.patrec.2020.03.004

104. Badea VE., Zamfiroiu A., Boncea R. Big Data in the Aerospace Industry. Informatica Economica. 2018; 22(1/2018): 17–24. Available at: DOI:10.12948/issn14531305/22.1.2018.02

105. Xia BS., Gong P. Review of business intelligence through data analysis. Benchmarking. 2014; 21(2): 300–311. Available at: DOI:10.1108/BIJ-08-2012-0050

106. Fleckenstein M., Fellows L. Modern data strategy. Modern Data Strategy. 2018. 1–263 p. Available at: DOI:10.1007/978-3-319-68993-7

107. Aleksandrova M. Big Data in the Banking Industry : The Main Challenges and Use Cases. https://easternpeak.com/blog/big-data-in-the-banking-industry-the-main-challenges-and-use-cases/ Accessed on 10/06/2019. 2019. pp. 1–9. Available at: DOI:url: https://easternpeak.com/blog/big-data-in-the-banking-industry-the-main-challenges-and-use-cases/

108. Institute S. Why Is Big Data Important? Whizlabs. 2015. Available at: DOI:http://www.sas.com/en_us/insights/big-data/what-is-big-data.html
(Accessed: 23 August 2021)

109. Raghupathi W., Raghupathi V. Big data analytics in healthcare: promise and potential. Health information science and systems. 2014; 2: 3. Available at: DOI:10.1186/2047-2501-2-3

110. Bumblauskas D., Gemmill D., Igou A., Anzengruber J. Smart Maintenance Decision Support Systems (SMDSS) based on corporate big data analytics. Expert Systems with Applications. 2017; 90: 303–317. Available at: DOI:10.1016/j.eswa.2017.08.025

111. Rengasamy D., Morvan HP., Figueredo GP. Deep Learning Approaches to

Aircraft Maintenance, Repair and Overhaul: A Review. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC. 2018; 2018-Novem: 150–156. Available at: DOI:10.1109/ITSC.2018.8569502

112. Hoyland T., Spafford C., Medland A. Mro Big Data – a Lion or a Lamb? Innovation and Adoption in Aviation Mro. 2016; : 16. Available at: http://www.oliverwyman.com/content/dam/oliver-wyman/global/en/2016/apr/NYC-MKT9202-001MRO-Survey-2016_web.pdf (Accessed: 21 August 2021)

113. Minadeo F. New Fleets Could Generate 98 Million Terabytes of Data Annually by 2026 According to Oliver Wyman's Annual MRO Survey. 2016.

114. Wang K., Wang Y. How AI Affects the Future Predictive Maintenance: A Primer of Deep Learning. Lecture Notes in Electrical Engineering. 2018. pp. 1–9. Available at: DOI:10.1007/978-981-10-5768-7_1

115. Bakshi K. Big data analytics approach for network core and edge applications. 2016 IEEE Aerosp. Conf. 2016; : 1–10. Available at: DOI:10.1109/AERO.2016.7500560

116. de Almeida AT., Cavalcante CAV., Alencar MH., Ferreira RJP., de Almeida-Filho AT., Garcez TV. Multicriteria and Multiobjective Models for Risk, Reliability and Maintenance Decision Analysis. 2015. Available at: DOI:10.1007/978-3-319-17969-8

117. Veluri S., Kumar R., Vasudevan R., Gorur RP., Nampuraja E., Shankaraiah M., et al. Improving Manufacturing Efficiencies through Industry 4.0 Technologies in Aerospace. SAE Technical Papers. 2018. pp. 1–7. Available at: DOI:10.4271/2018-01-1929

118. Alberto Fonte Silva Lima N., da Graça Tavares Álvares Serrão P., Martins Abrantes Leite A. Development of an Aircraft Health Monitoring Program for Predictive Maintenance Aerospace Engineering Examination Committee. 2017; (December): 118. Available at: https://fenix.tecnico.ulisboa.pt/downloadFile/1689244997258150/Thesis_73733

.pdf

119. Sethi C. Aerospace Bets on Big Data. 2015. Available at: DOI:https://www.asme.org/topics-resources/content/aerospace-bets-on-big-data

120. Satyanarayanan M., Simoens P., Xiao Y., Pillai P., Chen Z., Ha K., et al. Edge analytics in the internet of things. IEEE Pervasive Computing. 2015; 14(2): 24–31. Available at: DOI:10.1109/MPRV.2015.32

121. Weerasinghe S., Ahangama S. Predictive Maintenance and Performance Optimisation in Aircrafts using Data Analytics. 2018 3rd International Conference on Information Technology Research, ICITR 2018. IEEE; 2018; : 1–8. Available at: DOI:10.1109/ICITR.2018.8736157

122. Xu B., Kumar SA. Big Data Analytics Framework for System Health Monitoring. Proceedings - 2015 IEEE International Congress on Big Data, BigData Congress 2015. IEEE; 2015; : 401–408. Available at: DOI:10.1109/BigDataCongress.2015.66

123. Li S., Yang Y., Yang L., Su H., Zhang G., Wang J. Civil Aircraft Big Data Platform. Proceedings - IEEE 11th International Conference on Semantic Computing, ICSC 2017. IEEE; 2017; : 328–333. Available at: DOI:10.1109/ICSC.2017.51

124. Sciancalepore S., Piro G., Bruni F., Nasca E., Boggia G., Grieco LA. An IoT-based measurement system for aerial vehicles. 2nd IEEE International Workshop on Metrology for Aerospace, MetroAeroSpace 2015 - Proceedings. 2015; : 245–250. Available at: DOI:10.1109/MetroAeroSpace.2015.7180662

125. Ayhan S., Pesce J., Comitz P., Sweet D., Bliesner S., Gerberick G. Predictive analytics with aviation big data. Integrated Communications, Navigation and Surveillance Conference, ICNS. 2013; Available at: DOI:10.1109/ICNSurv.2013.6548556

126. Daily J., Peterson J. Predictive maintenance: How big data analysis can improve maintenance. Supply Chain Integration Challenges in Commercial Aerospace: A Comprehensive Perspective on the Aviation Value Chain. 2016. 267–278 p.

Available at: DOI:10.1007/978-3-319-46155-7_18

127. Nicchiotti G., Rüegg J. Data-Driven Prediction of Unscheduled Maintenance Replacements in a Fleet of Commercial Aircrafts. 2018; : 1–10. Available at: DOI:https://doi.org/10.36001/phme.2018.v4i1.237

128. Baum J., Laroque C., Oeser B., Skoogh A., Subramaniyan M. Applications of big data analytics and related technologies in maintenance-literature-based research. Machines. 2018; 6(4). Available at: DOI:10.3390/machines6040054

129. Eltoukhy AEE., Wang ZX., Chan FTS., Fu X. Data analytics in managing aircraft routing and maintenance staffing with price competition by a Stackelberg-Nash game model. Transportation Research Part E: Logistics and Transportation Review. 2019; 122(November 2018): 143–168. Available at: DOI:10.1016/j.tre.2018.12.002

130. Hiruta T., Uchida T., Yuda S., Umeda Y. A design method of data analytics process for condition based maintenance. CIRP Annals. CIRP; 2019; 68(1): 145–148. Available at: DOI:10.1016/j.cirp.2019.04.049

131. Puttini LC. IVHM development and the big data paradigm. SAE Technical Papers. 2013; 7. Available at: DOI:10.4271/2013-01-2332

132. Jha AK., Nayak S., Veerabhadrappa NK. an Architecture for Performing Real Time Integrated Health. 2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS). IEEE; 2017; : 1–5. Available at: DOI:10.1109/CSITSS.2017.8447679

133. Chen J., Lyu Z., Liu Y., Huang J., Zhang G., Wang J., et al. A big data analysis and application platform for civil aircraft health management. Proceedings - 2016 IEEE 2nd International Conference on Multimedia Big Data, BigMM 2016. IEEE; 2016; : 404–409. Available at: DOI:10.1109/BigMM.2016.54

134. Benedettini O., Baines TS., Lightfoot HW., Greenough RM. State-of-the-art in integrated vehicle health management. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering. 2009; 223(2): 157–170. Available at: DOI:10.1243/09544100JAERO446

135. Benedettini O., Baines TS., Lightfoot HW., Greenough RM. State-of-the-art in integrated vehicle health management. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering. 2009; 223(2): 157–170. Available at: DOI:10.1243/09544100JAERO446

136. Jennions IK., Niculita O., Esperon-Miguez M. Integrating IVHM and asset design. International Journal of Prognostics and Health Management. 2016; 7(2): 1–16. Available at: DOI:https://doi.org/10.36001/ijphm.2016.v7i2.2404

137. Aeronautics N. : GOALS AND OBJECTIVES FOR INTEGRATED VEHIC _ HEALTH MANAGEMENT (IVHM). 1992; Available at: DOI:https://www.hq.nasa.gov/office/aero/nra_pdf/ivhm_tech_plan_c1.pdf

138. Compare M., Baraldi P., Zio E. Challenges to IoT-Enabled Predictive Maintenance for Industry 4.0. IEEE Internet of Things Journal. IEEE; 2020; 7(5): 4585–4597. Available at: DOI:10.1109/JIOT.2019.2957029

139. Prepared N., Nra N. Survey of NASA V & V Processes / Methods. 2002; (April). Available at: DOI:https://sti.nasa.gov/

140. Poll S., Iverson D., Patterson-hine A. Characterization of model-based reasoning strategies for use in IVHM architectures. Nasa. 2001; Available at: DOI:https://doi.org/10.1117/12.487219

141. Mackey R., Iverson D., Pisanich G., Toberman M., Hicks K. Integrated System Health Management (ISHM) Technology Demonstration Project Final Report. 2015; Available at: DOI:https://sti.nasa.gov/ ISBN-13 : 978-1289242510

142. Sudolsky MD. IVHM solutions using commercially-available aircraft condition monitoring systems. IEEE Aerospace Conference Proceedings. 2007; : 1–8. Available at: DOI:10.1109/AERO.2007.352922

143. Jha BK., Ramachandra S., Srinivasa MPN. Conceptual Study on Integration of Engine Health Monitoring (EHM) System with Integrated Vehicle Health Monitoring (IVHM) System. INCOSE International Symposium. 2016; 26(s1): 55–69. Available at: DOI:10.1002/j.2334-5837.2016.00314.x

144. Engineering BT. Defense Technical Information Center Compilation Part Notice Bone Tissue Engineering. Materials Reseach Society Symposium Procedings. 2005; 635(July). Available at: DOI:https://apps.dtic.mil/sti/pdfs/ADP014232.pdf

145. Environments D. NASA SBIR 2014 Phase I Solicitation. NASA. 2014; : 1–2.

146. Byington CS., Roemer MJ., Kacprzymki GJ., Galie T. Prognostic enhancements to diagnostic systems for improved condition-based maintenance [military aircraft]. IEEE Aerospace Conference Proceedings. 2002; 6: 2815–2824. Available at: DOI:10.1109/AERO.2002.1036120

147. Keller K., Peck J., Swearingen K., Gilbertson D. Architectures for affordable health management. AIAA Infotech at Aerospace 2010. 2010; (April): 1–11. Available at: DOI:10.2514/6.2010-3435

148. Hess A., Fila L. The Joint Strike Fighter (JSF) PHM concept: Potential impact on aging aircraft problems. IEEE Aerospace Conference Proceedings. 2002; 6(August): 3021–3026. Available at: DOI:10.1109/AERO.2002.1036144

149. Li X., Wang H., Shen Y., Fu H. Integrated vehicle health management in the aviation field. Proceedings of 2016 Prognostics and System Health Management Conference, PHM-Chengdu 2016. IEEE; 2017; : 1–5. Available at: DOI:10.1109/PHM.2016.7819762

150. Khan F., Jennions I., Sreenuch T. Integration issues for vehicle level distributed diagnostic reasoners. SAE Technical Papers. 2013; 7. Available at: DOI:10.4271/2013-01-2294

151. Khoshafian S., Rostetter C. Digital Prescriptive Maintenance: Disrupting Manufacturing Value Streams through Internet of Things, Big Data, and Dynamic Case Management. Pega Manufacturing. 2020; : 1–20. Available at: DOI:https://www.pega.com/system/files/resources/2019-01/Digital-Prescriptive-Maintenance.pdf

152. Setrag K., Rostetter C. Digital Prescriptive Maintenance. Internet of Things, Process of Everything, BPM Everywhere. 2015; : 1-20. Available at: DOI:https://www.pega.com/industries/manufacturing/digital-prescriptive-

maintenance

153. Daigle MJ., Goebel K. Model-based prognostics with concurrent damage progression processes. IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans. 2013; 43(3): 535–546. Available at: DOI:10.1109/TSMCA.2012.2207109

154. Wu D., Jennings C., Terpenny J., Gao RX., Kumara S. A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests. Journal of Manufacturing Science and Engineering. 2017; 139(7): 071018. Available at: DOI:10.1115/1.4036350

155. Schwabacher M. A Survey of Data-Driven Prognostics. Infotech@Aerospace. 2005; (May). Available at: DOI:10.2514/6.2005-7002

156. Lopez-Martin M., Carro B., Sanchez-Esguevillas A. Application of deep reinforcement learning to intrusion detection for supervised problems. Expert Systems with Applications. Elsevier Ltd; 2020; 141: 112963. Available at: DOI:10.1016/j.eswa.2019.112963

157. Verhagen WJC., De Boer LWM. Predictive maintenance for aircraft components using proportional hazard models. J. Ind. Inf. Integr. Elsevier; 2018; (October 2017): 0–1. Available at: DOI:10.1016/j.jii.2018.04.004

158. Vianna WOL., Yoneyama T. Predictive Maintenance Optimization for Aircraft Redundant Systems Subjected to Multiple Wear Profiles. IEEE Systems Journal. IEEE; 2018; 12(2): 1170–1181. Available at: DOI:10.1109/JSYST.2017.2667232

159. Napierala K., Stefanowski J. Types of minority class examples and their influence on learning classifiers from imbalanced data. Journal of Intelligent Information Systems. Journal of Intelligent Information Systems; 2016; 46(3): 563–597. Available at: DOI:10.1007/s10844-015-0368-1

160. Błaszczyński J., Stefanowski J. Neighbourhood sampling in bagging for imbalanced data. Neurocomputing. Elsevier; 20 February 2015; 150(PB): 529–542. Available at: DOI:10.1016/j.neucom.2014.07.064 (Accessed: 13 September 2018)

161. Krawczyk B., Woźniak M., Herrera F. Weighted one-class classification for different types of minority class examples in imbalanced data. IEEE SSCI 2014 - 2014 IEEE Symposium Series on Computational Intelligence - CIDM 2014: 2014 IEEE Symposium on Computational Intelligence and Data Mining, Proceedings. 2015; : 337–344. Available at: DOI:10.1109/CIDM.2014.7008687

162. Hu Y., Guo D., Fan Z., Dong C., Huang Q., Xie S., et al. An Improved Algorithm for Imbalanced Data and Small Sample Size Classification. J. Data Anal. Inf. Process. 2015; 03(03): 27–33. Available at: DOI:10.4236/jdaip.2015.33004

163. Cieslak DA., Ryan Hoens T., Chawla N V., Philip Kegelmeyer W., Cieslak DA., Hoens TR., et al. Hellinger distance decision trees are robust and skew-insensitive. Data Min Knowl Disc. 2012; 24: 136–158. Available at: DOI:10.1007/s10618-011-0222-1

164. Charte F., Rivera AJ., Del Jesus MJ., Herrera F. MLSMOTE: Approaching imbalanced multilabel learning through synthetic instance generation. Knowledge-Based Systems. 2015; 89. Available at: DOI:10.1016/j.knosys.2015.07.019

165. Quintana E., Ibarra C., Escobedo L., Tentori M., Favela J. Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2012. 877–884 p. Available at: DOI:10.1007/978-3-642-33275-3

166. Torgo L., Ribeiro R. Predicting Rare Extreme Values. Springer, Berlin, Heidelberg; 2006. pp. 816–820. Available at: DOI:10.1007/11731139_95 (Accessed: 14 September 2018)

167. Torgo L., Branco P., Ribeiro RP., Pfahringer B. Resampling strategies for regression. Expert Systems. Wiley/Blackwell (10.1111); 1 June 2015; 32(3): 465–476. Available at: DOI:10.1111/exsy.12081 (Accessed: 14 September 2018)

168. Wang S., Minku LL., Yao X. A Systematic Study of Online Class Imbalance

Learning with Concept Drift. 2017; XX(X): 1–18. Available at: DOI:10.1109/TNNLS.2017.2771290

169. Nguyen HM., Cooper EW., Kamei K. Online learning from imbalanced data streams. SoCPaR. 2011; : 347–352. Available at: DOI:10.1109/SoCPaR.2011.6089268

CHAPTER 4: An Integrated Machine Learning Model for Aircraft Components Rare Failure Prognostics with Log-Based Dataset

This chapter details the proposed pre-processing strategy for dealing with a highly imbalanced log-based dataset. The chapter is formatted as a paper with the following content.

Predictive maintenance is increasingly advancing in the aerospace industry, and it comes with diverse prognostic health management solutions. This type of maintenance can unlock several benefits for aerospace organizations. Such as preventing unexpected equipment downtime and improving the service quality. One of the challenges that cause model performance degradation is the data-imbalanced distribution in developing data-driven predictive modelling. The extreme data imbalanced problem arises when the distribution of the classes present in the datasets is not uniform. Such that the total number of instances in a class far outnumber those of the other classes. Extremely skewed data distribution can lead to irregular patterns and trends, which affects the learning of temporal features. This paper proposes a hybrid machine learning approach that blends natural language processing techniques and ensemble learning for predicting extremely rare aircraft component failure. The proposed approach is tested using a real aircraft central maintenance system log-based dataset. The dataset is characterized by extremely rare occurrences of known unscheduled component replacements. The results suggest that the proposed approach outperformed the existing imbalanced and ensemble learning methods in terms of precision, recall, and f1-score. The proposed approach is approximately 10% better than the synthetic minority oversampling technique. It was also found that the class imbalance problem could be overcome by exclusively searching for patterns in the minority class. Hence, the model classification performance is improved.

4.1 Introduction

Airlines are increasingly concerned about the availability and reliability of assets and services. Most of them rely on scheduled maintenance to ensure that equipment is operating correctly in order to avoid unplanned breakdowns. Such types of maintenance are usually carried out on independent targeted components based on their usage without considering the relationship of components working together and influencing each other's lifetime. Moreover, this type of maintenance is labour-intensive and ineffective in identifying and predicting failures, especially in a complex system such as aircraft. In contrast, predictive maintenance helps identify anomalous behaviour from extensive historical failure data and turn it into meaningful, actionable insights for proactive maintenance – preventing downtime or accidents. This type of maintenance provides an intelligence forecast of when or if equipment will fail, so maintenance and repair can be scheduled before the failure occurs. Predictive maintenance requires working knowledge of the equipment, which can be achieved by installing sensors to record and monitor target variables. So that alerts are triggered when there is a violation of the defined threshold settings. This approach can sometimes be an effective solution in a simple system. However, it is impractical in a complex system since adding sensors to all components is unfeasible, especially in a large, cost-intensive fleet with potential regulatory challenges.

Furthermore, Fault detection, diagnostics, and prognostics (FDDP) have a huge potential to improve aircraft operational reliability and stability since the main aim of FDDP is to minimize losses while ensuring the safety of equipment and reducing the risk of unplanned breakdowns [1]. FDDP involves detecting the occurrence of fault as early as possible, classifying the fault type accurately, and predicting the next occurrences of such a fault. FDDP models are designed to detect anomalies of critical components by analyzing historical data to provide actionable alerts to the operators [2]. Since modern aircraft's operational and maintenance datasets have become much larger, the number of samples and dimensionality has increased. Therefore, implementing the traditional model-based and knowledge-based approaches are becoming too tricky [3].

Moreover, finding abnormal patterns in large log-based data is extremely challenging due to the complex non-linear relationships among the components process and sub-systems. Component failure resulting in unplanned breakdowns rarely occur during stable operation. The rare component failures create skewness or imbalanced distribution in the generated dataset [2], [3]. The imbalanced data problem has been shown to degrade data-driven models' performance, causing unreliable prognostics [4], [5]. The aforementioned challenges have motivated more research in the application of data-driven prognostics for conditioned-based maintenance in the aerospace industry [6].

In recent times, most of the aerospace industry's predictive maintenance are trends of modelling some specific data features such as vibration, pressure, exhaust gas, etc. More concentrated on the engine and auxiliary power unit [7], [8]. Whereas, considering predictive modelling at the system level is more efficient because the model will be able to capture the working relationship between components. Therefore, equipment failure logs are fruitful sources of information both for diagnoses and prognostics. However, intensive data pre-processing is required to harness valuable pieces of information. The recent technological advances have made equipment to operate through software applications. For example, modern aircraft are incorporated with advanced technology such as monitoring sensors and various aircraft communication systems (such as ACARS, ACMS) [9], which generates more extensive datasets. This application produces records of their operations, which includes some predefined parameters, failure messages, and other valuable target variables representing failures detected during the last operations—exploring such large historical record help in detecting an impending issue in advance. Therefore, relying on these flight failure records to develop predictive modelling for asset health management is a promising technique. The application of advanced analytics to anticipate maintenance needs to avoid the risk associated with service disruption [10].

Furthermore, building a predictive model from aircraft central maintenance system CMS data (which is the record of failure messages) in the total absence of digital sensor data measurements poses many challenges that are not yet fully explored. Many problems arise from learning with textual-based datasets. The first problem concerns the multidimensional data to be able to identify patterns leading to the component replacement. The second problem is mining patterns from random failure messages generated from different aircraft in a fleet [11]. The third problem is the inherent imbalanced distribution in the dataset. For instance, most of the failure messages in the CMS data are related to component replacement due to planned maintenance or not-fault-found. At the same time, the minority are related to unplanned component replacement (that is, the real unplanned replacements, which are our target in this study).

The imbalance classification problem or rare event occurrence is prevalent in many real-life application domains. For instance, detecting fraud in a credit card transaction, where most transactions are legitimate and few are fraudulent. The fraudulent minority transactions are more important to predict than the legitimate majority because the consequences can be grave if any fraudulent transaction goes unnoticed [12]. Likewise, most patients can be healthy in clinical diagnosis, while a few diagnose a certain rare disease [13]. The costs of misclassifying infected as healthy cannot be tolerated because of the high risks of deterioration and fatality.

Similarly, Imbalanced classification can also be applied in aircraft predictive maintenance modelling, where most of the generated failure messages represent false alarm or no fault found, and the minority represents the real faults that resulted in component replacement. The problem of imbalanced data in aircraft predictive maintenance modelling using log-based CMS failure messages is that component failure rarely occurs, which creates imbalanced distribution in the generated dataset. In some cases, the ratio between classes in the dataset can be as high as 10000:1, which is known as an extreme imbalanced problem [14]–[16]. Moreover, identifying patterns and learning from extreme imbalanced datasets increases classification challenges in machine learning. Hence, an improved method of accurately recognizing the minority class instances is a required process [17], [18].

Several research approaches have been conducted to solve the imbalanced classification problem. The imbalanced classification problem's solution can be categorized into three main groups; the data level, the algorithm level, and the hybrid approach see Figure 4-1. The data level approach involves resampling the dataset before presenting it as an input to the learning algorithm. The algorithm level approach tackles the imbalanced data problem by modifying the traditional machine learning algorithms to respond favourably to both classes during learning [19]. The hybrid method combines two or more algorithms or data level approaches to achieve better performance.

Although imbalance classification problems have been extensively researched [20], [21], the open literature lacks an exhaustive unified solution to generally handle the problem for predictive modelling. Hence, it is still an open area of research. Therefore, this study aims at developing an actionable prognostics model, which will enable the anticipation of unscheduled maintenance activities relating to aircraft functional items replacements, which can be achieved by identifying predictive signatures in the CMS failure messages.

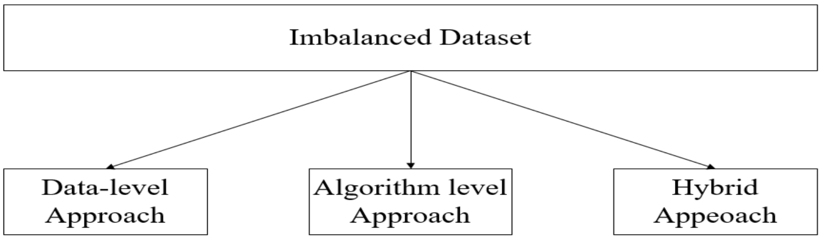


Figure 4- 1 Three approaches to handling the imbalanced dataset problem

Firstly, we propose a novel hybrid model that blends natural language processing techniques and ensemble learning for predicting rare aircraft component failure using imbalanced textual log-based data. The model is based on a log-based pattern identification technique, which involves transforming and integrating well-known natural language processing techniques (the TF-IDF and Word2vec) and ensemble learning for pattern identification and classification. It uses log-based aircraft central maintenance system data, which is not often used for predictive maintenance modelling. In addition, our approach helps in tackling the extreme imbalanced classification problem by searching for patterns exclusively in the minority class, which improves model performance. In predictive maintenance, a state-of-the-art ensemble-learning algorithm is adapted as a base classifier; we also show how unscheduled maintenance can be mitigated using reliable and robust prognostic models.

This paper is structured as follows; in section 2, we present related work and a description of the datasets. Section 3 explains the methodology and the proposed approach. Section 4 presents the case study and experimental setup and discuss the result. Finally, conclusions and future work are presented in section 5.

4.2 Related Work

The system's operational logs are well studied in different application domains [22]. Each application domain has its specific requirements that have an impact on the design and development of the corresponding solution. Some researchers have focused on log data for troubleshooting and anomaly detection solutions [23], [24]. Other application domains that have practically shown its use are computer hard-disk failure prediction [25], [26], medical equipment failure [27], [28], and many more [29]. System failure messages obtained from logs can also be used to understand the equipment's behaviours and common failure patterns. The most closely related work to our approach is failure-messages-based machine learning modelling. Li et al. [30] provide an approach for mining system log files. The authors attempt to understand and categorise common failure patterns that resulted in system failures.

In recent times Natural Language Processing has been shown to help extract useful information from text data [31][53]. NLP is a field of Artificial Intelligence (AI) that studies the interaction between human language and machines, mainly how to program computers to process and analyse large amounts of natural language data. NLP has widely been used to solve text classification, summarisation, extraction problems. Different techniques can be used to assign text into categories according to the content. There are various available methods for extracting raw text data, such as a bag of words (using term-frequency inverse frequency), word embedding [54]. The application of NLP is seen in many domains. For example, Tanguy et al. [31] show the application of NLP in mining text-based aviation incident reports data to identify future threats. However, they did not further show how such failure can be predicted to prevent its future occurrence[55]. Po-Hao Chen et al. [56] use the combination of NLP algorithms the term frequency-inverse document frequency (TF-IDF) and term frequency weighting (TF) to Categorize oncologic response in radiology reports. The open literature lacks a study that has transformed natural language techniques such as TF-IDF and Word vectorization method for pattern identification using the ACMS dataset.

The use of aircraft data has been used for modelling. For example, Korvesis et al. [32], use aircraft post-flight report data to develop an event failure prediction system via multi-instance regression. In contrast, we focus on classification and finding a solution to the rare occurrence of failures instead of regression. Another close work that uses the aircraft ACMS dataset to develop the predictive model is Nicchiotti et al. [9]. The authors design a two-step process, a transformed Eigen-face from image processing to produce the signatures of the different types of maintenance action; a Support Vector Machine (SVM) is used to select the flight legs candidate for a prognostics alert. Our study explores the impact of extremely rare failure on the predictive model by the following strategy. Natural language processing techniques were used to identify patterns related to aircraft component failures. Likewise, Yan et al. [33] proposed a predictive model to predict faults with high priority in advance by exploring the historical data of aircraft maintenance systems. The authors did not take into consideration the problem of the rare failure occurrence, which is part of our focus in this study. Their study also considers single aircraft instead of a fleet. Analysing fleet data can be more challenging; therefore, our methodology considers a fleet-based approach instead of a single aircraft. Verhagen et al. [34] develop an approach to reduce unscheduled maintenance by focusing on identifying operational factors affecting component reliability. The research uses a statistical data-driven approach and, the authors applied a proportional hazard model on aircraft operational and maintenance datasets. Though, their work uses an aircraft operational dataset, which is closely related to the one used in our study. However, because of the multi-variate nature of the ACMS data, our approach focuses on exploring the application of machine learning.

Another related category for predicting failure from log-based data is the rule-based expert system [35], [36]. In this type of approach, preconditioned rules are defined; the rules are then matched against the input data. If the predefined condition is met, a failure alert will be triggered. The rules are mainly defined by domain experts, not through data mining. Vilalta et al. [37] described the use of a rule-based approach to detect patterns in the sequence of events. In practice, rule-based approaches are more effective for a small and simple system. Its application in a large and complex system is quite challenging and, in some cases, impractical because domain experts need to continually update the rules in the event of any upgrades or changes, which is cumbersome.

Another related category for processing log-based data is the application of sequential pattern mining [38], [39], which is mainly about extracting interesting, useful, and unexpected patterns across sequential data using a statistical approach. Many studies have shown the applicability of using text sequence mining for failure prediction in a complex system [40]–[42]. We explore the sequence mining techniques and find out that applying sequence pattern mining alone is not suitable for our problem because of the rare occurrence of unplanned aircraft component replacement.

Although there are many existing approaches in the literature, some are suitable for solving failure prediction in specific types of equipment. Hence, the particularities of our data (Heterogeneous in nature containing symbolic sequences, numeric time-series, categorical variables and unstructured text. It is a non-trivial task to translate free-text log messages into meaningful features) limit us from using an out-of-the shelf approach. Our approach differs from the aforementioned approaches in many aspects. We proposed a new approach of pre-processing the aircraft central maintenance log-based data. In addition, the new approach provides a solution to the imbalanced classification problem, which enhances the model performance. Finally, the proposed hybrid machine learning technique for aircraft component replacement prediction is developed.

Our approach applies a unique combination of TF-IDF and Word2Vec notions from the NLP to extract patterns and categorise failure messages into common failures. Considering text-based aircraft CMS failure messages, each segment of patterns is considered a document, and each pattern is considered a word. TF-IDF helps in pruning out unproductive and redundant patterns. At the same time, Word2Vec is used to find the most relevant documents related to the target component. It also helps in converting words into a vector of numbers. The approach improves the categorization accuracy by considering CMS failure messages' temporal characteristics, which improves the overall performance of the predictive model (such as reducing false positives, increasing prediction recall, and precision). The approach also includes a solution to the rare occurrence of target failures by searching for patterns exclusively in the minority class.

4.3 Methodological Approach

This section describes the methodology used in this study.

As seen in Figure 4-2, the traditional machine learning framework is divided into three phases: The pre-processing phase, the model training, the testing and validation phase, and the model deployment phase. Building a machine-learning model from any data source must often deal with imperfect data. Therefore, we ensure data quality by cleaning the data - correcting outliers, handling missing values, and aggregating impossible combinations before further analysis. Cleaning and transforming the data is necessary because jumping into analyzing data that has not been carefully screened for such problems can produce misleading results [43].

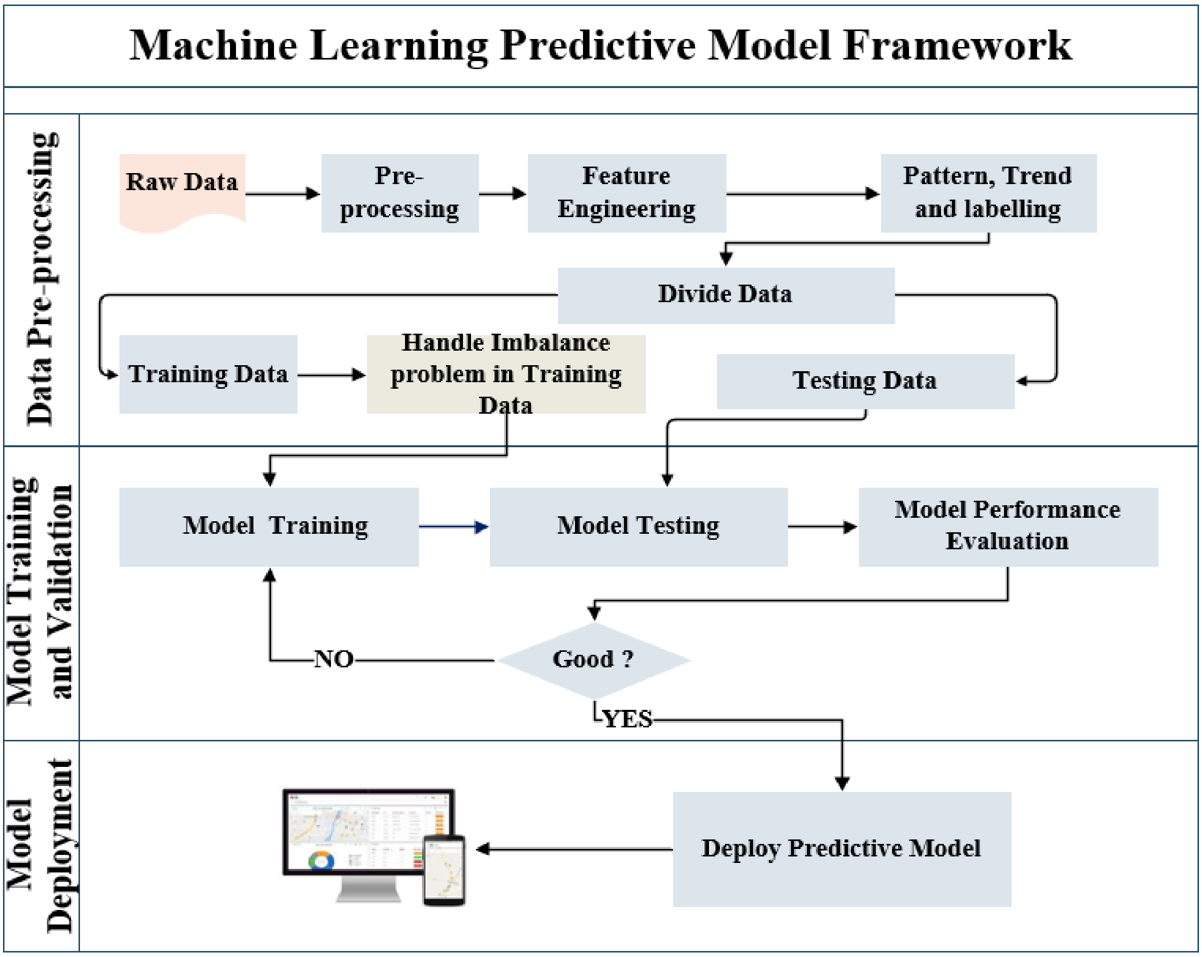


Figure 4- 2 The pipeline for developing a predictive model using an imbalanced dataset

Secondly, the feature engineering step is necessary to select the best predictors for our problem. The datasets are merged at this stage. The feature engineering process determines the right features that best describe our target components. The aircraft operational log data contains timestamp and flight cycle numbers, making it easier for creating windows. The third step involves identifying component failure patterns and trends. We focus on the component replacement that occurs due to unplanned maintenance. The aim is to find the best framework for processing time-series, log-based datasets with rare failures, focusing on addressing the imbalanced classification problem and improving the base learning algorithm's performance.

4.3.1 Problem Description

We formally describe the log-based rare failure prediction problem as follows. Given a functional item number FIN of a particular aircraft family A, with the rare occurrence of failure. Using log-based failure messages $fI_m(A)$ Collected from a fleet. Can we infer the probability of its replacement $\pi_R(T)$ within a time window T? This problem can be solved using machine learning; hence, we consider it as a binary and multi-class classification problem for predicting aircraft functional item failure with a given period T. The training data contains predictive features extracted from the log of failure messages obtained from a civil aircraft fleet. The failure labels are provided from the actual aircraft maintenance record. Note that our solution is targeted at specific functional items replacements, not a generic replacement. In addition, the targeted functional items are extremely rare, and our goal is to develop a model that can overcome the challenge of rarity while making predictions. The prognostic system's main aim is to adequately provide failure alerts early enough to give maintenance engineers enough time to deal with the problem before it actually occurs. Also, the alerts should not come too early to avoid component wastage due to premature replacement. Therefore, the prediction window needs to be defined using domain expert knowledge. A prediction window is defined as; at least two flights and no more than ten flights in advance in this study. The dataset is imbalanced; hence the imbalanced problem is defined as follows: Slightly imbalanced is when the imbalanced ratio (IR) between classes is approximately 5% to 30%. If IR is less than 5%, we consider it to be an extremely imbalanced problem. Finally, our prediction aim is, for each selected Functional Item Number (FIN), we target to achieve at least 50% prediction of unscheduled maintenance

4.3.2 The proposed approach

This section discusses the implementation of our novel approach. The approach can be applied in multivariate time series, text-based, and imbalanced datasets. Therefore, the raw aircraft ACMS data is sequential and in time-series format. The flight cycles are also in sequence. The failure messages are text-based. The records of unplanned components replacements are rare in the dataset. This specification makes it suitable to test our approach. We also focus on solving the extreme imbalanced problem to enhance the reliability and performance of data-driven models. Thus, improving predictive models will mitigate the risk associated with unscheduled maintenance.

We use natural language processing and time series analysis approaches to find trends and patterns. Natural language processing -Term Frequency-Inverse Document Frequency (TF-IDF) and the word2vec approach are used to recognise target component patterns and trends, as shown in Figure 4-3. We made several assumptions to account for the infrequent incidence of component substitution in the flight dataset. For example, the replacement of aircraft components is described by categorical and text-based characteristics and occurs at irregular intervals. Secondly, we also assume that target components are less represented (highly infrequent). Therefore, we develop our algorithm to exclusively search for all patterns preceding each target component to predict the next replacement. To achieve that, we transform the TF-IDF and word2vec technique to evaluate the importance of each failure or error message [33], [44]. TF-IDF is a machine learning Natural Language Processing (NLP) word embedding technique that weighs words in text mining [45]. This technique allows us to represent text in a coordinated system where related error messages are placed closer together based on the corpus of relationships. It helps us also to filter out unrelated failure messages. The TF-IDF consists of two parts, namely,

1. TF- Term Frequency: which calculates the frequency of word appearance in a document. If a given term is t

$$\therefore \; \boldsymbol{TF(t)} = \frac{\boldsymbol{n_t}}{\boldsymbol{n_d}} \qquad (4\text{- }1)$$

Where n_t is the total number of times t appear in a document and n_d is the total number of terms in the document.

2. IDF- Inverse Document Frequency: which measures the importance of each term in the document.

$$\therefore\ IDF(t) = log\ \frac{m_d}{m_t} \qquad (4\text{- }2)$$

Where, m_t is the total number of documents that contain term t and m_d is the total number of documents

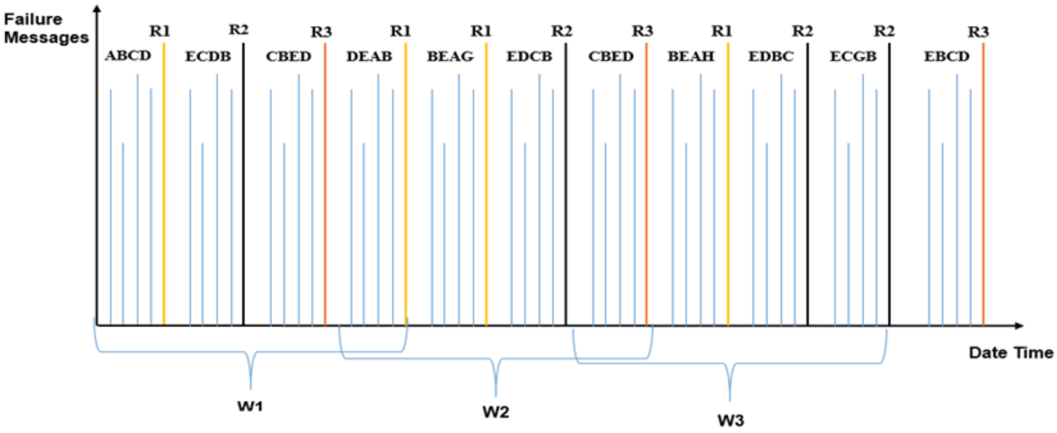


Figure 4- 3 Failure message patterns- A, B, C... represents CMS failures messages and R1, R2... represents LRU replacements

Putting it all together

$$\Rightarrow TF\_IDF(t) = TF(t).IDF(t) \qquad (4\text{- }3)$$

Therefore, implementing the above approach, we denote document to be each window in the dataset, and term t to be target components, and failure messages represent words. For instance, for a given dataset, let R_1 be the first component replaced due to unplanned breakdown of equipment, R_2 for second replacement and R_3 for third and so on. Let the alphabet (A, B, C, etc.) represent the failure messages. Therefore, all failure messages in a window preceding each replacement constitute a pattern and are represented as follows.

R_1 → ABC, YZP, PPB….

R_2 → XYZ, AEP, CDB….

R_3 → CDA, EDM, OPN….

We then identify the pattern for each $\boldsymbol{R_i}$ within that window. For instance, looking at Figure 4-3, W1 = {R_1: (ABEG), R_2: (ECDB), R_3: (GBED), R_2: (DEAB)}. We then find all patterns that are related to each target component replacement $\boldsymbol{R}$ across all the datasets. Finally, the extracted patterns are then used to train the algorithm.

Therefore, during model training, taken, for example, all failure messages related to R_1 are identified and all the possible combinations of failure messages related to R_1 are created, which produces more new different patterns. This is done to increase more patterns related to each replacement, which will address the imbalanced problem. The combination and creation of new patterns are achieved using bootstrapping techniques. We use the select with a replacement approach to avoid the overfitting problem.

Furthermore, the model is developed to flag up component replacement prognostic alerts when a pattern is detected. The features, such as date-time and flight cycle numbers, play a vital role in defining when the model should flag up prognostic alerts in advance.

Furthermore, our pattern recognition strategy is similar to the one developed by Vilalta et al. [37]. However, our approach differs in the learning strategy instead of using the rule-based model; we make use of supervised learning (classification technique) to build a data-driven model. The imbalanced classification problem is overcome by exclusively searching for patterns in the minority class. The strategy is shown in Algorithm 1. Having known the patterns, the next step is we represent the features into a vector space using the word2vec method. Prior to that, categorical features are handled using the one-hot-encoding technique [46]. As shown in Figure 4-3, all the terms in the pattern are selected and then converted into a vector space dimension. To illustrate, considering the windows w_i and patterns ABC ... leading to components replacement$s$ F_i.

W_1= ABEGR_1 –CBDER_3 –DEABR_1 –EDCBR_2

W_2= AEDBR_1 –BEAGR_1 –CDCBR_2-DEBCR_3

W_3= EDCBR_2 –ABEGR_1 –EDBCR_2 –CBDER_3

Using TF-IDF and word2vec approach to identify all failure messages related to each target component replacement.

$$Boolean\ frequencies = m(t) \begin{cases} 1,\ if\ t\ occurs\ in\ window \\ 0,\ otherwise \end{cases} \qquad (4\text{- }4)$$

Where $m(t)$ represents the term frequency of patterns of failure messages leading to each component replacements. All corresponding replacements in each window are then sum up. tf (t,w) is the total number of patterns present in each window.

The inverse document frequency measures the total number of each pattern in relation to components replaced in each window. That is if it is common or it is rare across all windows.

$$Idf\ (t, W) = log\ \frac{N}{|\{d \in W : t \in d\}|} \quad (4\text{- }5)$$

Where N is the total number of failure messages in a window N = $|W|$, and $|\{d\ \in W: t\ \in d\}|$, a number of windows where the term t appears.

$$tf\_idf(t, d, W)\ =\ td(t, d). idf(t, W) \quad (4\text{- }6)$$

We then create our feature victor using the following equation.

$$\overrightarrow{vw_n} = tf(t_1, w_n), tf(t_2, w_n), tf(t_3, w_n), \ldots, tf(t_n, w_n) \quad (4\text{- }7)$$

Using a continuous bag of words strategy in word vectorization, each dimension of the feature vector is represented by the pattern, for example, $tf(t_1, w_n)$ represents the frequency of term 1. For example, using equation 4-4 and Table 4-1, patterns for window 1 to 3 are represented as victors as follows:

$$\overrightarrow{vw_1} = tf(t_1, w_1), tf(t_2, w_1), tf(t_3, w_1), tf(t_4, w_1), \ldots, tf(t_n, w_1) \quad (4\text{- }8)$$

$$\overrightarrow{vw_1} = (2,1,1 \ldots \mathrm{n})$$

$$\overrightarrow{vw_2} = tf(t_1, w_2), tf(t_2, w_2), tf(t_3, w_2), tf(t_4, w_2), \ldots, tf(t_n, w_2) \quad (4\text{- }9)$$

$$\overrightarrow{vw_2} = (2,1,1 \ldots \mathrm{n})$$

$$\overrightarrow{vw_3} = tf(t_1, w_2), tf(t_2, w_3), tf(t_3, w_3), tf(t_4, w_3), \ldots, tf(t_n, w_3) \quad (4\text{- }10)$$

$$\overrightarrow{vw_3} = (1,1,1 \ldots \mathrm{n})$$

Table 4- 1 Sample of the pre-processed aircraft CMS dataset

Date	Time	Flight circle	A/C No	Window lag	FM pattern	FIN Rplmt
10-03-15	09.03	-91	1	W_1	ABEG	R_1
10-03-15	10.03	-88	2	W_1	DEAB	R_1
11-03-15	10.00	-81	8	W_1	EDCB	R_2
11-03-15	11.05	-80	21	W_1	CBED	R_3
13-04-15	09.08	-79	12	W_2	AEDB	R_1
13-04-15	10.03	-76	9	W_2	BEAG	R_1
14-04-15	22.00	-73	23	W_2	EDCB	R_2
15-04-15	09.05	-71	2	W_2	CBED	R_3
16-04-15	09.02	-70	3	W_3	BEAH	R_1
16-04-15	21.08	-65	18	W_3	ABCG	R_3
17-04-15	13.00	-64	28	W_3	EDBC	R_2

The resulting vectors show that window one $\overrightarrow{vw_1} = (2,1,1\ldots)$ has two patterns that prompt replacement of the component R_1 , one pattern for R_2, and one for R_3 We then represent it in a general matric with the shape $|w| * l$ where $|w|$ is the cardinality of the feature vector space in each window, and l is the total number of pattern vectors. The unique patterns are then encoded and tranformed into feature space.

$$M = \begin{vmatrix} 2 & 1 & 1\ldots\mathrm{n} \\ 2 & 1 & 1\ldots\mathrm{n} \\ 1 & 1 & 1\ldots\mathrm{n} \\ 1 & 2 & 0\ldots\mathrm{n} \end{vmatrix}$$

Algorithm 1: Detecting patterns and trends of target components

- Find the pattern of failure message preceding the target component within a given fixed window.
- Carry out validation of characters that uniquely identify the target component.
- Combine the characteristic to build a data-driven predictive model.

The pseudocode:

```
INPUT:
 Imbalanced time series dataset
{
F = Sequence of failure messages
fm =failure message
W = window size
r = Target replacements
T= Time
}
OUTPUT:  P = Pattern for Target Replacement
TARGET_PATTERN ( F, W, r, T)
        Step 1: Get the Data D
                Initialize variables G= 0, H=0
                Define window size W
        Step 3: loop through the series of event in each W to identify a component
                 replacement.
                FOREACH  F,  $f_{m(i)}$ = ( $r_i, t_i$ ) ∈ $F$
                 (where $t_i = current\ time$)
        Step 4:  Identify a pattern preceding the component replacement.
                        FOREACH F,  $f_{m(j)}$ = ( $r_j, t_j$ ) ∈ $H$
                                If ($current\ time - t_j$) > W ; Remove $f_{m(i)}$ from H
                        END
        Step 5: Generate a pattern for each event that occurs together, leading to the
                 replacement of the component.
                        IF $f_{m(i)}$ is a target replacement
                                G = G ∪ {$r_j$| $r_{j,}$ ...}
                                H ∪ $f_{m(i)}$
                        Identify frequent patterns [51] on G
                END
        Step 6: Next window: Go-to step 3
                Use TF-IDF on G to find all related pattern M
        Step 7: Output M
```

Algorithm 1 transverses through the sequence of failure messages, which is in time-series format. The algorithm stores patterns of failure messages related to each target functional item failure in memory. The identified patterns are then used for fault prediction.

Validation: To validate the performance of predicting aircraft components failure from imbalanced log-based data with the proposed approach, we modelled it into binary classification and multi-class classification. In the first scenario, we modelled it as a multi-class classification problem that predicts all the targeted component failures at the same time. Secondly, we modelled it as a binary classification problem that predicts individual functional items. In both instances, we use ensemble-learning algorithms as a base-classifier. We choose to evaluate the approach using random forest ensemble learning because of its capability of combining more than once classifiers to achieve better results, which has an advantage over a single classifier, especially in a skewed data distribution context. To evaluate the model in terms of imbalance classification, we compare our proposed approach with the existing synthetic minority oversampling technique (SMOTE).

As shown in Table 4-1, the data is grouped into two categories representing different types of aircraft in the fleet. The A330 –long-range (LR) and the A320 -Single-aisle (SA) aircraft. The dataset ranging from 2011 to 2015 is used to train the model, while from 2016 to 2018 is used for testing. After the pattern identification-using algorithm 1, the resulting dataset is divided into two (for training and testing). Data ranging from 2011 to 2015 is used for model training, while from 2016 to 2018 is used for evaluation and testing.

The effectiveness of the proposed approach was demonstrated on the log-based CMS dataset. We choose a target functional item Number (FIN) of high practical value for each aircraft family with an adequate number of known failure cases. We selected out of many the following aircraft functional items to be used in the experiment. The target components selected for this study are based on some group of common failures in an aircraft subsystem that happens with a frequency of 0.1 - 1% over some time.

LRU for A330 –long-range (LR) aircraft family: 4000KS - Electronic Control Unit/ Electronic Engine Unit, 4001HA – Pressure Regulating Valve, 5RV1 – Satellite Data unit, and 438HC – Trim Air Valve.

LRU for A320 -Single-aisle (SA) aircraft family: 11HB – Flow control valve, 10HQ - Avionics equipment ventilation computer, 1TX1 - Air traffic service unit, and 8HB - Flow control valve 2.

Imbalanced Ratio (IR): In the A330 aircraft family, the size of the training dataset is 360575, and the A320 family size is 389829. The frequency of functional items replacement emanating from unscheduled maintenance is as follows. In the A330 aircraft family, 4001HA is replaced 17 times, giving us the imbalance ratio (IR) of 360558: 17, 4000KS is replaced 15 times given us IR of 360560: 15, 5RV1 is replaced 16 times given us IR of 360559: 16, and 438HC is replaced 25 times given us IR of 360550: 25. Similarly, in the A320 aircraft family, 11HB is replaced 11 times, giving us the imbalance ratio (IR) of 389818: 11, 10HQ is replaced 12 times given us IR of 389817: 12, 1TX1 is replaced 25 times given us IR of 389804: 25, and 8HB is replaced 14 times given us IR of 389815: 14.

$$\mathbf{IR} = \frac{Minority\ class}{Majority\ class} * 100$$

(4- 11)

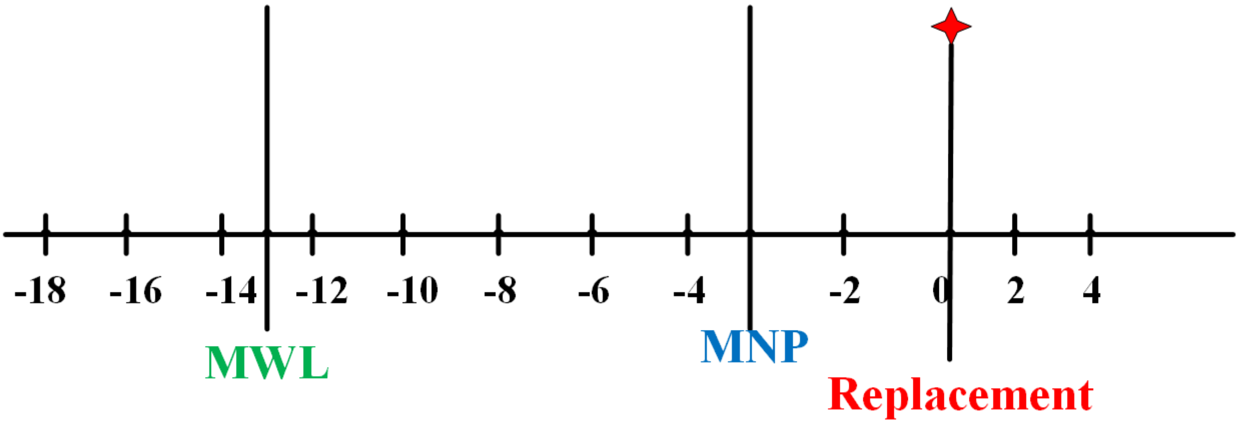


Figure 4- 4 Representation of flight cycles from replacement

Scenario 1: multi-class approach

We make a prediction for all FIN and compare it against the baseline imbalanced learning algorithm -SMOTE.

1. SMOTE + Random Forest (RF) After cleaning the data. We divided the data into training and testing. The training data was resampled using SMOTE. Then the different machine learning algorithms are used to train the classifier.

2. Our approach + Random Forest (RF), After cleaning the data. We carry out behavioural pattern analysis. We then divided the data into training and testing. We train the model without applying any existing imbalanced learning method. Then the different machine learning algorithms are used to train the classifier.

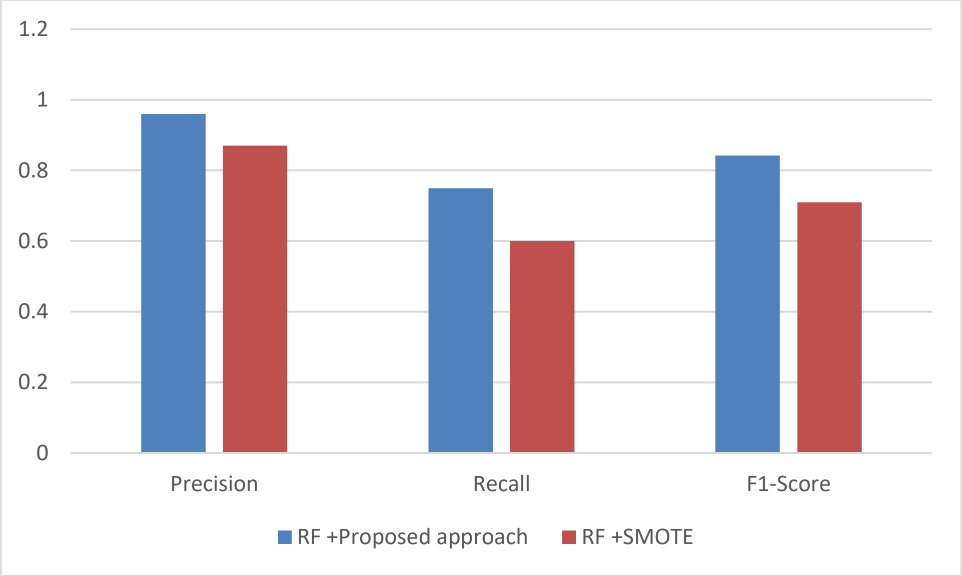


Figure 4- 5 Showing the performance comparison between SMOTE and the proposed method using the random forest as a classifier in both cases

In the first instance, we consider all the aforementioned targeted FIN failures. During evaluation, accuracy, recall, and precision is used as performance metrics. The comparison result of the two cases is shown in Figure 4-7. Random forest outperformed other ensemble classifiers. Therefore, in the second scenario, which is predicting individual functional items (binary classification approach), we use only a random forest algorithm.

Scenario 2: Binary classification approach- Individual component failure prediction model:

We make a prediction for each FIN and compare it against the baseline imbalanced learning algorithm -SMOTE.

Any machine-learning algorithm for classification can be used in choosing the base-classifier for binary classification. Our choice of an ensemble-learning algorithm as a base-classifier is effective in improving predictive performance, especially in classifying skew datasets. In addition, because RF is an ensemble bagging technique that combines multiple decision trees to achieve a better result. The trees in RF create high variance and low bias, making it a suitable choice. Also, since data is distributed over different trees in the forest, and each tree sees a different set of data, in general, RF does not over-fit. Because they are made of low bias trees, it does not suffer from the under-fitting problem. Thus, we choose a random forest among the ensemble algorithm because it gives better precision and recall compared to others. We use algorithm 1 to generate patterns related to each targeted FIN. We then adept the RF as seen in Figure 4-8

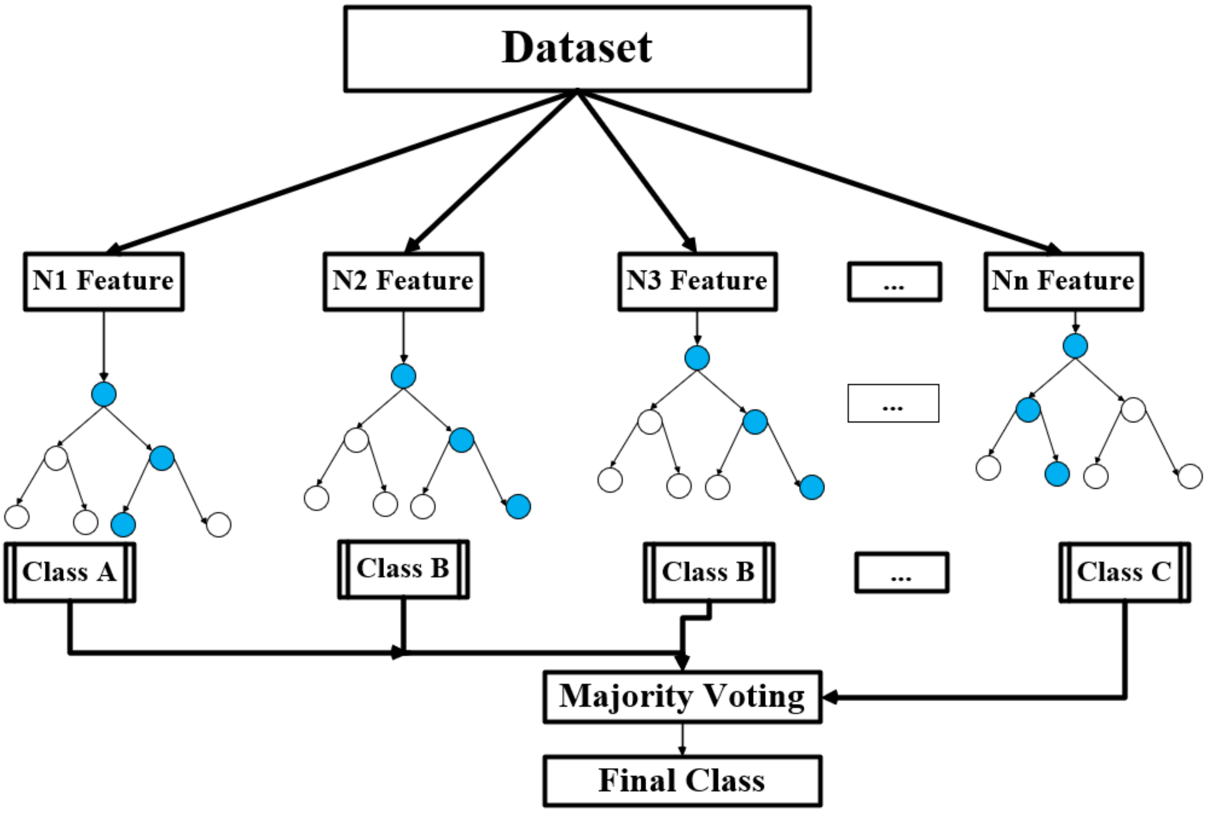


Figure 4- 6 Random Forest Ensemble Algorithm

Algorithm to create the individual failure prognostic model. RF is an ensemble learning method where the training data is divided into several subsamples, and each subsample is trained using a decision tree classifier know as a weaker learner. The result is then aggregated by majority voting providing a stronger base learning algorithm. Apart from sampling on the dataset, trees are randomized by using boosting and bagging techniques to generate splits [49], [50]. This approach enhances the performance of the model.

In predicting targeted individual functional items, their failures are extremely rare. Normally, accuracy is mostly considered an important metric to evaluate the performance of a classifier. However, the use of accuracy to evaluate performance under extreme imbalanced problems can be misleading because classifies will be biased towards the majority class to achieve high overall accuracy. Therefore, to evaluate the classifiers' performance more precisely, some alternative metrics are adapted, which include precision, recall, F1-score, and area under the curve.

4.5 Result and Discussion

As shown in Figures 4-7 , it is observed that comparing our approach with SMOTE using different ensemble learning algorithms as base-classifier. The performance of all the base-classifiers is better with the proposed approach compared to SMOTE. Furthermore, RF outperformed other ensemble algorithms; it shows comparative performance in recall and precision, which means RF can identify more faults than other base-classifiers. Although the multi-class approach produced a significant improvement, most predictions fall close to the defined maximum wasted life.

As shown in Table 4-2, For individual FIN prediction. It can be observed that our model has a precision of more than 70% for all the functional items. It means whenever the model predicts aircraft failure, that leads to component replacement.

Table 4- 2 Showing experiment results using binary classification approach with RF as the base classifier

A330 Aircraft											
		RF+ SMOTE				RF + Our approach					
IR	FIN	Precision	Recall	F1	AUC	Precision	Recall	F1	AUC	TPR	FPR
0.0047	4001HA	0.83	0.62	0.70	0.72	0.94	0.79	0.86	0.87	0.79	0.21
0.0043	4000KS	0.80	0.60	0.68	0.69	0.90	0.76	0.82	0.83	0.76	0.24
0.0044	5RV1	0.80	0.60	0.68	0.69	0.91	0.77	0.83	0.84	0.77	0.23
0.0069	438HC	0.90	0.85	0.87	0.88	0.96	0.85	0.84	0.86	0.85	0.15
A320 Aircraft											
0.0028	11HB	0.70	0.59	0.64	0.65	0.81	0.70	0.75	0.76	0.70	0.30
0.0031	10HQ	0.75	0.62	0.68	0.67	0.86	0.72	0.78	0.79	0.72	0.28
0.0064	1TX1	0.88	0.80	0.83	0.84	0.91	0.82	0.86	0.87	0.82	0.18
0.0036	8HB	0.80	0.66	0.72	0.73	0.88	0.74	0.80	0.81	0.74	0.26

In other words, this indicates that out of the total prediction, the model prognoses more than seventy percent of failures that lead to LRU replacement. The precision score also shows the model produces less than thirty percent of false-positive alerts. Similarly, an average recall of more than 60% is achieved in all the considered FIN's. Indicating that the model correctly predicts more than sixty percent of actual failure that leads to LRU replacement. It is important to note that for individual prediction (binary classification), most prediction falls close to the defined minimum notice period, which means the component will be adequately utilized. This means binary classification has an advantage over multi-class prediction. Since the high cost associated with false-negative is the main concern in this study- that is a misclassifying real failure as not failure, especially for safety-critical equipment where the consequence is grave. Therefore, the recall score shows that the model triggers 60% of the actual failure alert, leading to LRU replacement.

The goal is to obtain both a high percentage of precision and recall in all cases. However, more than 20% of the false positives and 30% false-negative rates are still recorded. Nevertheless, our approach achieved our target: to predict more than 50% of aircraft component replacement within the desired defined range (in-between MNP and MWL). This can be seen by the overall percentage F1-score, which is approximately 65% in all cases. Similarly, to obtain the trade-off between the model sensitivity (TPR) and specificity (1-FPR), ROC Curves of each target component replaced is acquired.

The graphical representation of the average result obtained is presented in Figure 4-10 to 4 - 13; as seen in most of the cases, the area under the curve (AUC for the testing dataset is above 70%. Indicating good overall sensitivity of the classifier to predicting component replacement due to unscheduled maintenance). Note that the ROC curve does not depend on data distribution. This makes it useful in evaluating classifiers predicting imbalanced datasets.

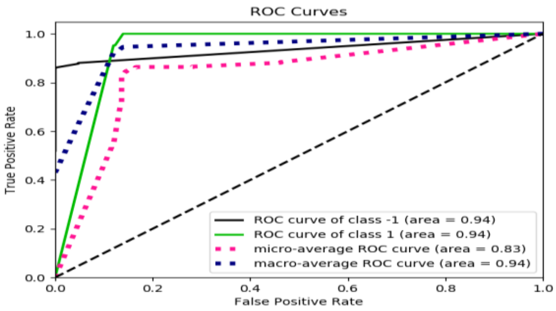


Figure 4-9. ROC for 4001HA

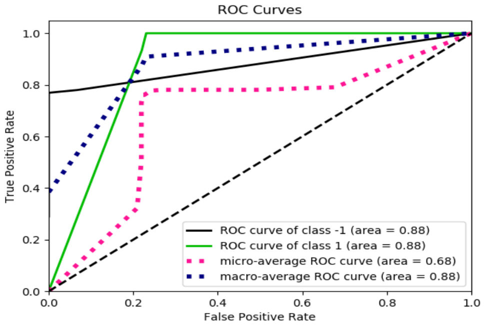


Figure 4-10. ROC for 4000KS

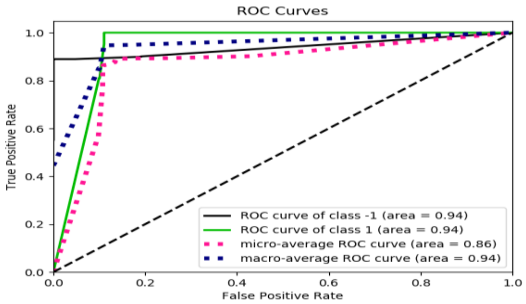


Figure 4-11. ROC for 5RV1

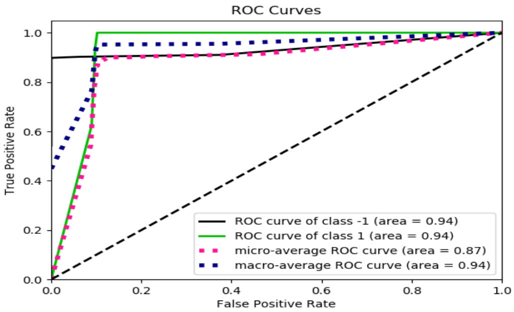


Figure 4-12. ROC for 438HC

Furthermore, although the proposed approach achieved approximately 20% of the overall percentage of the false-positive rate, in contrast, SMOTE achieved an approximately overall false-positive rate of 30%. This shows a difference of 10%, indicating that our approach achieved a significant improvement compared to synthetic minority oversampling techniques. Furthermore, it can be observed that the imbalanced ratio has an impact on performance. For instance, in extreme IR cases, we obtain a lower precision and recall compared to the ones with higher IR. Despite the extreme imbalance ratio in all the cases considered, our approach still achieved better performance than SMOTE, which indicates its robustness in handling extreme imbalanced datasets.

4.6 Conclusion

This paper proposes an integrated data-driven learning technique for predicting aircraft component failure using imbalanced, textual, and log-based data. A hybrid model involves blending natural language processing techniques, and ensemble prediction is developed to tackle extreme imbalanced classification problem and forecast aircraft component failures. We utilize real-life aircraft Central Maintenance System (CMS) data to develop a predictive maintenance model for predicting aircraft component replacement in advance to avoid unscheduled maintenance. A well-known natural language processing technique, the TF-IDF and Word2vec, are transformed for pattern identification and text vectorization. Then an ensemble random forest algorithm was successfully adapted for individual functional item prediction. In predictive maintenance, we show how unscheduled maintenance can be mitigated using the proposed robust prognostic model. The model can flag off component replacement alerts within the desired define range. In evaluation, we suggest an evaluation criterion that combines the prognostics alerts with the precision and recall within a reasonable timeframe. We compare the performance of our proposed approach against state-of-the-art imbalanced learning techniques (SMOTE) in terms of F1 score. The proposed approach is approximately 10% better than SMOTE. It was also found that the class imbalance problem can be overcome by searching for patterns in the minority class exclusively. Hence, the model classification performance is improved. Finally, even though the proposed method can predict more than 50% of unscheduled aircraft component failure, it did not go further to determine the root cause of the failure. Therefore, this work can be extended to enhancing aircraft failure diagnosis using proactive logging data. Future work will also aim to increase the model's performance by exploiting information from a variety of sources, such as sensors and other related variables.

4.7 Acknowledgement

This work has been performed under the AIRMES (Airline Maintenance Operations implementation of an E2E Maintenance Service Architecture and its enablers) project. This project has received funding from the Clean Sky 2 Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 681858. This project has also received funding from the Petroleum Technology Development Fund (PTDF) of Nigeria.

4.8 Reference

1. Dai X., Gao Z. From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis. IEEE Transactions on Industrial Informatics. IEEE; 2013; 9(4): 2226–2238. Available at: DOI:10.1109/TII.2013.2243743

2. Saufi SR., Ahmad ZA Bin., Leong MS., Lim MH. Challenges and Opportunities of Deep Learning Models for Machinery Fault Detection and Diagnosis: A Review. IEEE Access. IEEE; 2019; 7: 122644–122662. Available at: DOI:10.1109/access.2019.2938227

3. Park P., Di Marco P., Shin H., Bang J. Fault detection and diagnosis using combined autoencoder and long short-term memory network. Sensors (Switzerland). 2019; 19(21): 1–17. Available at: DOI:10.3390/s19214612

4. Raghuwanshi BS., Shukla S. UnderBagging based reduced Kernelized weighted extreme learning machine for class imbalance learning. Engineering Applications of Artificial Intelligence. Elsevier Ltd; 2018; 74(July): 252–270. Available at: DOI:10.1016/j.engappai.2018.07.002

5. Wu Z., Lin W., Ji Y. An Integrated Ensemble Learning Model for Imbalanced Fault Diagnostics and Prognostics. IEEE Access. 2018; 6: 8394–8402. Available at: DOI:10.1109/ACCESS.2018.2807121

6. Jinsong B., Yuan G., Xiaohu Z., Jianguo Z., Xia J. A Data Driven Model for Predicting Tool Health Condition in High Speed Milling of Titanium Plates Using Real-Time SCADA. Procedia CIRP. The Author(s); 2017; 61: 317–322. Available at: DOI:10.1016/j.procir.2016.11.191

7. Nicchiotti G., Rüegg J. Data-Driven Prediction of Unscheduled Maintenance Replacements in a Fleet of Commercial Aircrafts. 2014; : 1–10. Available at: DOI:https://doi.org/10.36001/phme.2018.v4i1.237

8. Austin J., Jackson T., Fletcher M., Jessop M., Cowley P., Lobner P. Predictive Maintenance : Distri buted Ai rcraft Engi ne Diagnostics. 2004;

9. Nicchiotti G., Rüegg J. Data-Driven Prediction of Unscheduled Maintenance Replacements in a Fleet of Commercial Aircrafts. 2018; : 1–10. Available at: DOI:https://doi.org/10.36001/phme.2018.v4i1.237

10. Oster C V., Strong JS., Zorn CK. Analyzing aviation safety: Problems, challenges, opportunities. Research in Transportation Economics. Elsevier Ltd; 2013; 43(1):

148–164. Available at: DOI:10.1016/j.retrec.2012.12.001

11. Alestra S., Bordry C., Brand C., Burnaev E., Erofeev P., Papanov A., et al. Rare event anticipation and degradation trending for aircraft predictive maintenance. 11th World Congress on Computational Mechanics, WCCM 2014, 5th European Conference on Computational Mechanics, ECCM 2014 and 6th European Conference on Computational Fluid Dynamics, ECFD 2014. 2014. pp. 6571–6582.

12. Nghiem LT. MASI : Moving to Adaptive Samples in Imbalanced Credit Card Dataset for Classification. 2018 IEEE International Conference on Innovative Research and Development (ICIRD). IEEE; 2018; (May): 1–5.

13. Gao T., Hao Y., Zhang H., Hu L., Li H., Li H., et al. Predicting pathological response to neoadjuvant chemotherapy in breast cancer patients based on imbalanced clinical data. Personal and Ubiquitous Computing. 2018; : 1–9. Available at: DOI:10.1007/s00779-018-1144-3

14. Janjua ZH., Vecchio M., Antonini M., Antonelli F. IRESE: An intelligent rare-event detection system using unsupervised learning on the IoT edge. Engineering Applications of Artificial Intelligence. Elsevier Ltd; 2019; 84(September 2018): 41–50. Available at: DOI:10.1016/j.engappai.2019.05.011

15. He H., Garcia EA. Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. 2009; 21(9): 1263–1284. Available at: DOI:10.1109/TKDE.2008.239

16. Elrahman SMA., Abraham A. A Review of Class Imbalance Problem. Netw. Innov. Comput. 2013; 1: 332–340. Available at: DOI:www.mirlabs.net/jnic/index.html

17. Olmo JL., Cano A., Romero JR., Ventura S. Binary and multiclass imbalanced classification using multi-objective ant programming. International Conference on Intelligent Systems Design and Applications, ISDA. 2012; : 70–76. Available at: DOI:10.1109/ISDA.2012.6416515

18. Buda M., Maki A., Mazurowski MA. A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks. 2018; 106: 249–259. Available at: DOI:10.1016/j.neunet.2018.07.011

19. Haixiang G., Yijing L., Shang J., Mingyun G., Yuanyue H., Bing G. Learning from class-imbalanced data: Review of methods and applications. Expert Systems with

Applications. 2017; 73: 220–239. Available at: DOI:10.1016/j.eswa.2016.12.035

20. Branco P., Torgo L., Ribeiro RP. A Survey of Predictive Modelling under Imbalanced Distributions Adapting Resampling Strategies for Dependency-Oriented Data in Imbalanced Domains View project International Workshop on Cost-Sensitive Learning View project A Survey of Predictive Modelling . 2015. Available at: https://www.researchgate.net/publication/275968092

21. Bi J., Zhang C. An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme. Knowledge-Based Systems. 2018; 158: 81–93. Available at: DOI:10.1016/j.knosys.2018.05.037

22. Salfner F., Lenk M., Malek M. A survey of online failure prediction methods. ACM Computing Surveys. 2010; 42(3): 1–68. Available at: DOI:10.1145/1670679.1670680

23. Gorinevsky D., Matthews B., Martin R. Aircraft anomaly detection using performance models trained on fleet data. Proceedings - 2012 Conference on Intelligent Data Understanding, CIDU 2012. IEEE; 2012; : 17–23. Available at: DOI:10.1109/CIDU.2012.6382196

24. Ge Z., Song Z., Ding SX., Huang B. Data Mining and Analytics in the Process Industry: The Role of Machine Learning. IEEE Access. 2017; 5: 20590–20616. Available at: DOI:10.1109/ACCESS.2017.2756872

25. Murray JF., Hughes GF., Kreutz-Delgado K. Machine learning methods for predicting failures in hard drives: A multiple-instance application. Journal of Machine Learning Research. 2005; 6: 783–816.

26. Son J., Zhou Q., Zhou S., Mao X., Salman M. Evaluation and comparison of mixed effects model based prognosis for hard failure. IEEE Transactions on Reliability. IEEE; 2013; 62(2): 379–394. Available at: DOI:10.1109/TR.2013.2259205

27. Sipos R., Fradkin D., Moerchen F., Wang Z. Log-based predictive maintenance. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2014. pp. 1867–1876. Available at: DOI:10.1145/2623330.2623340

28. Yuan Y., Zhou S., Sievenpiper C., Mannar K., Zheng Y. Event log modeling and

analysis for system failure prediction. IIE Transactions (Institute of Industrial Engineers). 2011; 43(9): 647–660. Available at: DOI:10.1080/0740817X.2010.546385

29. Zhang K., Xu J., Min MR., Jiang G., Pelechrinis K., Zhang H. Automated IT system failure prediction: A deep learning approach. Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016. IEEE; 2016; : 1291–1300. Available at: DOI:10.1109/BigData.2016.7840733

30. Li T., Ma S., Liang F., Peng W. An integrated framework on mining logs files for computing system management. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2005; : 776–781. Available at: DOI:10.1145/1081870.1081972

31. Tanguy L., Tulechki N., Urieli A., Hermann E., Raynal C. Natural language processing for aviation safety reports: From classification to interactive analysis. Computers in Industry. 2016; 78: 80–95. Available at: DOI:10.1016/j.compind.2015.09.005

32. Korvesis P., Besseau S., Vazirgiannis M. Predictive maintenance in aviation: Failure prediction from post-flight reports. Proceedings - IEEE 34th International Conference on Data Engineering, ICDE 2018. IEEE; 2018; : 1423–1434. Available at: DOI:10.1109/ICDE.2018.00160

33. Yan W., Zhou J-H. Predictive modeling of aircraft systems failure using term frequency-inverse document frequency and random forest. IEEE International Conference on Industrial Engineering and Engineering Management. 2018. pp. 828–831. Available at: DOI:10.1109/IEEM.2017.8290007

34. Verhagen WJC., De Boer LWM. Predictive maintenance for aircraft components using proportional hazard models. J. Ind. Inf. Integr. Elsevier; 2018; 12(October 2017): 23–30. Available at: DOI:10.1016/j.jii.2018.04.004

35. Butler KL. Expert system based framework for an incipient failure detection and predictive maintenance system. Proceedings of the International Conference on Intelligent Systems Applications to Power Systems, ISAP. 1996; : 321–326. Available at: DOI:10.1109/isap.1996.501092

36. Kumar P., Srivastava RK. An expert system for predictive maintenance of mining excavators and its various forms in open cast mining. 2012 1st International

Conference on Recent Advances in Information Technology, RAIT-2012. IEEE; 2012; : 658–661. Available at: DOI:10.1109/RAIT.2012.6194607

37. Vilalta R., Sheng Ma. Predicting rare events in temporal domains. 2002 IEEE Int. Conf. Data Mining, 2002. Proceedings. 2003; : 474–481. Available at: DOI:10.1109/icdm.2002.1183991

38. Abbasghorbani S., Tavoli R. Survey on sequential pattern mining algorithms. Conference Proceedings of 2015 2nd International Conference on Knowledge-Based Engineering and Innovation, KBEI 2015. 2016; : 1153–1164. Available at: DOI:10.1109/KBEI.2015.7436211

39. Truong-Chi T., Fournier-Viger P. A Survey of High Utility Sequential Pattern Mining. 2017; 1(1): 97–129. Available at: DOI:10.1007/978-3-030-04921-8_4

40. Fu X., Ren R., Mckee SA., Zhan J., Sun N. Digging deeper into cluster system logs for failure prediction and root cause diagnosis. 2014 IEEE International Conference on Cluster Computing, CLUSTER 2014. 2014; (2): 103–112. Available at: DOI:10.1109/CLUSTER.2014.6968768

41. Chang W., Xu Z., You M., Zhou S., Xiao Y., Cheng Y. A Bayesian failure prediction network based on text sequence mining and clustering. Entropy. 2018; 20(12). Available at: DOI:10.3390/e20120923

42. Lim HK., Kim Y., Kim MK. Failure Prediction Using Sequential Pattern Mining in the Wire Bonding Process. IEEE Transactions on Semiconductor Manufacturing. IEEE; 2017; 30(3): 285–292. Available at: DOI:10.1109/TSM.2017.2721820

43. Shichao Z., Chengqi Z., Qiang Y. Data Preparation for Data Mining. Applied Artificial Intelligence. 2007. 37–41 p. Available at: DOI:10.1080/08839510390219264

44. Dubin D. The Most Influential Paper Gerard Salton Never Wrote. Library Trends. 2004; 52(4): 748–764.

45. Kamath U., Liu J., Whitaker J. Deep Learning for NLP and Speech Recognition. 2019. Available at: DOI:10.1007/978-3-030-14596-5

46. Cerda P., Varoquaux G., Kégl B. Similarity encoding for learning with dirty categorical variables. Machine Learning. 2018; 107(8–10): 1477–1494. Available at: DOI:10.1007/s10994-018-5724-2

47. Airbus. ACMS Discription Manual. Airbus; 2000. Available at: DOI:https://wenku.baidu.com/view/179923f4910ef12d2af9e723.html

48. Manuals A training. Trouble shooting philosophy with A320 CFDS / A340 CMS. 2005; (May).

49. Gutschi C., Furian N., Suschnigg J., Neubacher D., Voessner S. Log-based predictive maintenance in discrete parts manufacturing. Procedia CIRP. Elsevier B.V.; 2019; 79: 528–533. Available at: DOI:10.1016/j.procir.2019.02.098

50. Inoue H., Inoue R. A very large platform for floating offshore facilities. Coastal ocean space utilization III. Proc. symposium, Genoa, 1993. 1995; : 533–551.

51. Agrawal R. Fast Algorithms For Mining Association Rules In Datamining. International Journal of Scientific & Technology Research. 2013; 2(12): 13–24.

53. Chen L., Song L., Shao Y., Li D., Ding K. Using natural language processing to extract clinically useful information from Chinese electronic medical records. International Journal of Medical Informatics. Elsevier; 2019; 124(December 2018): 6–12. Available at: DOI:10.1016/j.ijmedinf.2019.01.004

54. Christian H., Agus MP., Suhartono D. Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF). ComTech: Computer, Mathematics and Engineering Applications. 2016; 7(4): 285. Available at: DOI:10.21512/comtech.v7i4.3746

55. Grosz BJ. Natural language processing. Artificial Intelligence. 1982; 19(2): 131–136. Available at: DOI:10.1016/0004-3702(82)90032-7

56. Chen PH., Zafar H., Galperin-Aizenberg M., Cook T. Integrating Natural Language Processing and Machine Learning Algorithms to Categorize Oncologic Response in Radiology Reports. Journal of Digital Imaging. Journal of Digital Imaging; 2018; 31(2): 178–184. Available at: DOI:10.1007/s10278-017-0027-x

CHAPTER 5: Proposed Ensemble and Hybrid Learning Techniques for Imbalanced Dataset

This chapter presents new proposed techniques for handling extreme imbalanced classification problems based on ensemble-hybrid learning using heterogeneous datasets. Two new methods are proposed and implemented: Balanced Calibrated Hybrid Ensemble Technique (BACHE) algorithm and a hybrid soft mixed Gaussian processes with the expectation-maximization (EM) algorithm. The documentation of the proposed algorithm is presented as follows:

5.1 Handling Imbalanced Data for Aircraft Predictive Maintenance using the BACHE Algorithm

Developing a prognostic model to predict an asset's health condition is a maintenance strategy that increases asset availability and reliability through better maintenance scheduling. Therefore, developing reliable vehicle health predictive models is vital in the aerospace industry, especially considering a safety-critical system such as aircraft. However, one of the significant challenges faced in building reliable data-driven prognostic models is the imbalance dataset. Training machine-learning models using an imbalanced dataset causes classifiers to be biased towards the class with majority samples, resulting in poor predictive accuracy in data-driven models. This problem can become more challenging if the imbalance ratio is extreme. This paper develops a novel approach called Balanced Calibrated Hybrid Ensemble Technique (BACHE) to tackle the severe imbalanced classification problem. The proposed method involves the combination of hybrid data sampling and ensemble-based learning. It uses a cascading balanced approach to transfer a class imbalance problem into a sub-problem by decomposing the original problem into a set of subproblems, each characterized by a reduced imbalance ratio. Then uses a calibrated boosting with a cost-sensitive decision tree to enhance recognition of hard to learn patterns, which improves the prediction of the extreme minority class. BACHE is evaluated using a real-world aircraft dataset with rare component replacement instances. Also, a comparative experiment of the proposed approach with other similar existing methods is conducted. The performance metrics used are precision, recall, G-mean, and an area under the curve. The final results show that the proposed model outperforms other similar methods. Also, it can attain an excellent performance on large, extremely imbalanced datasets.

5.1.1 Introduction

The technological growth in the aerospace industry and the continued advancement in data analytics have made the generation and analysis of large quantities of aircraft data more affordable. Therefore, this has caused a transformation in maintenance strategies by shifting from preventive maintenance to predictive maintenance. Research into the development of data-driven prognostic models for condition-based maintenance is gaining more attention [1,2]. However, researchers' major problems are the low representation of faulty asset behaviour, which results in an imbalanced dataset. The imbalanced data problem arises when the distribution of classes present in the dataset is not uniform, such that the total number of instances in one class far outnumber that of the other class [3]. The imbalance problem degrades the performance of the data-driven model, causing imprecise prognostics. The rapid flow of data from the industrial process has brought about an increasing research focus in big data analytics and its many applications in academics, industries, and government sectors [2,4,5]. Therefore, solving the imbalanced classification problem is necessary in order to build a high-performance predictive model. Research into this area is still an open issue [6–8], especially the data-driven approaches [9].

The imbalanced classification problem is prevalent in many application domains. For Example, in building predictive maintenance for aircraft, the historical data is often imbalanced because the record about systems and processes is mostly healthy with fewer failure records [10]. Similarly, in financial fraud detection, in most cases, illegal transactions are often rare compared to the majority of legitimate ones. The fraudulent minority transactions are more critical to predict accurately to avoid the consequence of the successful occurrence of fraud [11]. The application of imbalanced learning is also seen in clinical science for rare disease detection. The majority of the population is healthy, and the minority is infected [12]. In this case, predicting the minority becomes critical. Likewise, imbalance learning can be seen in the oil spillage detecting problem. Large images of an ocean captured by satellite may show a few images representing the oil spillage portion, and most of the images represent the non-spillage areas [13]. In such cases, the target is to predict the minority spillage portion of the ocean.

In a situation where the ratio between classes is not significantly large, and the existing machine learning methods can adequately handle such an imbalanced problem. However, in a situation where the ratio between classes in the dataset is extreme, say, 10000:1 [14], learning becomes more challenging because examples from the overwhelming class can be well-classified, whereas samples from the minority class can be misclassified. In the worst case, minority examples are treated as outliers or noise of the majority class and ignored or dropped during learning. The learning algorithm ends up generating a trivial classifier that classifies every example as the majority class. Other factors that can impact the classification algorithm's performance apart from the imbalance ratio are; the class's small disjunct, the noise, and the class overlapping[15,16].

In this study, we consider the problem of an imbalanced dataset in the context of aircraft' predictive maintenance.

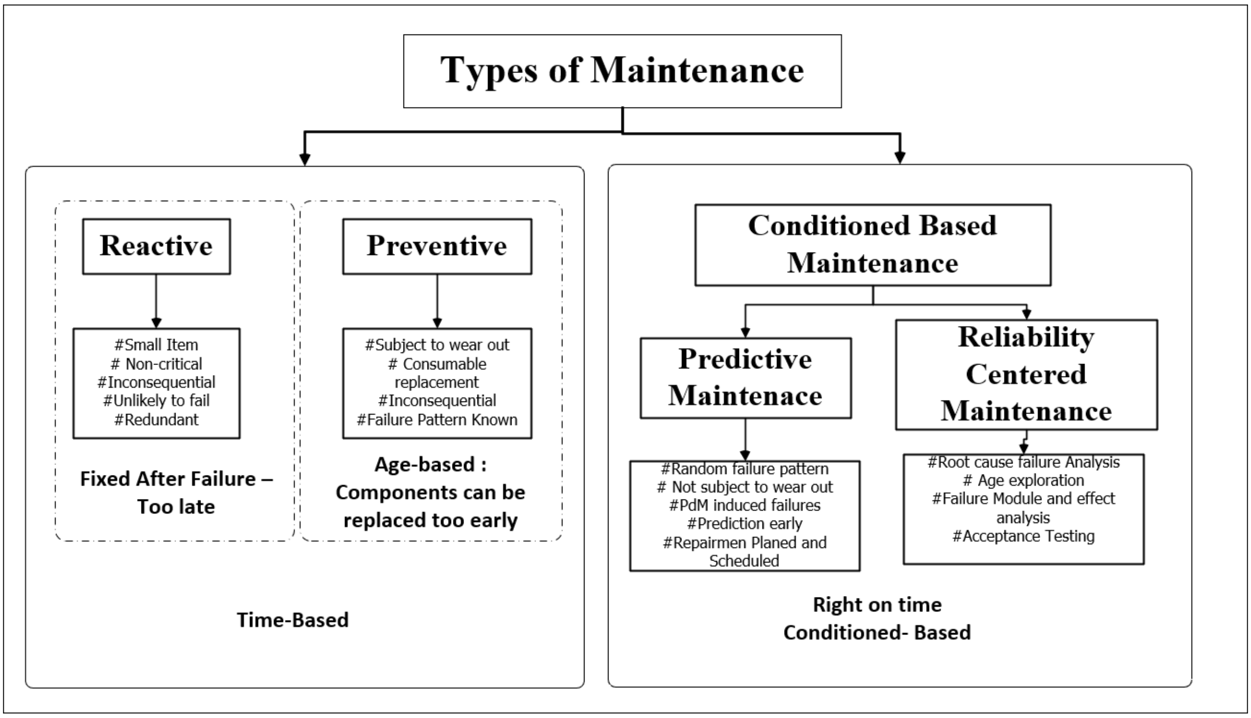


Figure 5- 1 Types of Maintenance strategies

Maintenance strategies have progressively developed over time, and the goal remains the same that is to preserve equipment. Recently, industries are becoming more aware of the advantages of applying advanced machine learning methods to enhance quality, process performance, and system uptime by maintaining overall equipment effectiveness (OEE). This development has brought more research attention to predictive maintenance modelling for heavy equipment monitoring [5,9,17].

Maintenance can be categorized into two types: a time-based and conditioned base, as seen in Figure 5-1. In time-based, we have reactive maintenance, which involves fixing after things have been broken down, and preventive maintenance involves keeping things from breaking. Reactive maintenance is quite expensive and time-consuming because no prior knowledge is available to plan effective maintenance. On the other hand, preventive maintenance allowed the pre-emptive measure to be taken before equipment failure, for example, by conducting repairs at fixed intervals regardless of the equipment condition. Advancement in technology has allowed the second category of maintenance, known as conditioned-based maintenance (CBM). CBM optimizes preventive activities based on the actual conditions of the asset. Predictive maintenance is a form of CBM where a predictive model is developed to forecast future failure using past failure records.

This study considers the imbalanced dataset's problem in developing a data-driven prognostic model for predicting unplanned aircraft component failures. The study proposes a novel method that involves a unique fusion of two machine learning techniques (ensemble learning and cost-sensitive learning) to form a hybrid approach. In the proposed hybrid algorithm, we use a balance-cascading algorithm to cascade the majority class. Then the minority class is synthesized and boosted using a data expansion policy, which overcomes the extreme imbalance classification problems and reduces the computational cost for larger datasets compared to deep learning methods [18]. The ensemble process provides a unique classifier arrangement and cost sensitivity to each weak learner, which produces state-of-the-art performance.

The contribution of this paper is as follows:

One of the fundamental research questions that this implementation seeks to answer is can a class overlapping and small disjunct problem inherent in the extremely imbalanced ACMS dataset be overcome by using hybrid ensemble learning? A new algorithm known as Balanced Calibrated Hybrid Ensemble Technique (BACHE) is designed and implemented to answer the above question. The approach's novelty is found in the uniqueness of ensemble architecture that combines weak classifiers, which balanced the bias-variance tradeoff to improve minority class sample prediction. Cost-sensitive are defined in each weak classifier to improve the prediction of minority class samples. Another contribution is that the proposed approach's effectiveness is validated using a real-world dataset (the aircraft central maintenance system-CMS dataset). This is a distinctive contribution because of the dataset's heterogeneous nature, which is challenging to mine for predictive modelling.

The remainder of this paper is organized as follows: Section 5.1.2 provides related work. Section 5.1.3 presents the methodology. Section 5.1.4 discusses the results and model validation, and finally, we present our conclusion and future work in Section 5.1.5.

5.1.2 Related Work

This section gives an overview of imbalanced classification problems. Several research approaches have been conducted to solve the imbalanced classification problem, and some comprehensive reviews can be found in [19–21]. The solution to the imbalanced classification problem can be categorized into three main groups [20]. The data level, the algorithm level, and the hybrid approach, as seen in Figure 5-2. The data level approach involves resampling the dataset before presenting it as an input to the learning algorithm. The data level approach has gained a lot of research attention, especially the over-sampling techniques, which involve increasing the minority class samples to have a balanced class. Some of the methods are based on oversampling are the Synthetic Over-sampling Techniques (SMOTE) developed by Nitesh et al. [22]. Their technique creates new synthetic samples into the minority class to balance with the majority. Though, the SMOTE approach has widely addressed the imbalanced classification problem. However, SMOTE contains some drawbacks, such as class overlapping because it ignores adjacent samples when creating new synthetic points [23] and an overgeneralization problem. Hence, many advanced versions of SMOTE have been developed, such as SMOTE-Boost [24], which introduces new dynamic weighted synthetic data points in the minority class at each round of boosting steps to eliminate the overgeneralization problem. SMOTE-boost tries to solve these drawbacks of the main SMOTE by adding synthetic data points in each weak classifier of the easy-ensemble method [25]. Other versions are the Easy-SMOTE Algorithm [26], Borderline-SMOTE [27], and many more.

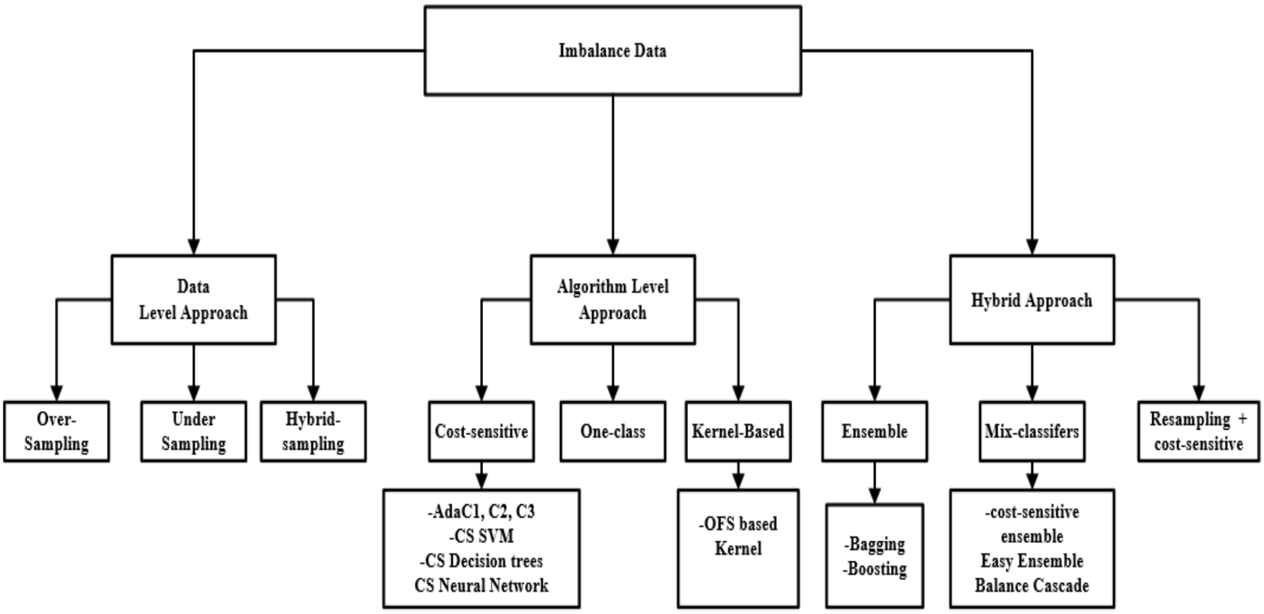


Figure 5- 2 The three existing categories of the State-of-the-art approach of the handling imbalance problem

The algorithm level approach tackles the imbalanced learning problem by altering the learning algorithm to respond favourably to both classes during learning [20]. Cost-sensitive learning is an algorithm-level approach. The cost-sensitive method is explored by defining the cost of misclassification for each class. Determining the cost of misclassification is challenging in the traditional classification algorithms (such as support vector machines, decision trees, and more) because the algorithms presume that all classification errors carry the same cost. Hence, they focus on minimizing the error rate and the percentage of a class's incorrect prediction, ignoring the difference between the misclassification errors. Therefore, cost-sensitive learning takes into consideration the different costs that vary by type of classification (true-positive, true-negative, false-positive, false-negative) across all samples. The goal is to minimize the total cost, such as the G-mean score. From a business point of view, it is vital to determine the misclassification cost for each class. For instance, in aircraft maintenance, the cost of misclassifying the minority class (failure) has a higher impact on the business than the misclassifying majority class (healthy state). Hence, cost-sensitive learning is used to mitigate such problems [28]. Cost-Sensitive in Decision Tree algorithm (CS-DT) involves introducing cost into the decision tree algorithm for the algorithm to respond favourably to all classes during training[29]. Cost-sensitive learning is effective in classifying datasets with different imbalance distributions [28]. The changing misclassification costs are best understood using the idea of a cost matrix. As seen in Table 5-1, a Cost sensitives learning can be binary or multi-class; in either case, it associates different misclassification costs to every prediction.

Table 5- 1 Cost or Confusion Matrix

	Actual Positives	Actual Negatives	TP: True Positive TN: True Negative FP: False Positives FN: False Negative
Predicted Positives	TP ($C_{1,1}$)	FP ($C_{1,-1}$)	
Predicted Negatives	FN ($C_{-1,1}$)	TN ($C_{-1,-1}$)	

Using the confusion matrix as shown in Table 5-1, the value ($C_{i,j}$) represents the cost of misclassifying a data point from its actual class (j) to a predicted class (i), 1 represents positive class, while -1 represents the negative class. Usually, the cost of correct prediction that is TP and TN should always be lower than the cost of misclassification error that is FN and FP, usually is set to zero. ($C_{i,i}$) is regarded as a negated error since the data point is predicted correctly. Cost-sensitive learning has widely been applied in imbalanced learning [28]; the challenge is learning the cost matrix. In some domains, it might be obvious because the consequence of misclassification can just be based on monetary value. However, in areas such as predictive maintenance for aircraft, the consequence of the misclassification of faults can be grave.

The easiest way of defining the misclassification cost is to input it manually according to the domain expert advice or inversely calculate it based on class distribution [30–32]. The challenge of using a manual approach for calculating the cost of misclassification is that it is time-consuming and sometimes impractical. Another approach can be to fit the importance of features to adaptive equations [33], which involves incorporating second-order information to enhance the prediction of the minority class. However, because of the peculiarity of the dataset used in this study, neither method is suitable. Hence, we define the misclassification cost from cost-sensitive algorithms' evaluation functions, using weighted Platt calibration to measure the cost sensitivity of the classification algorithm.

The imbalanced learning hybrid approach involves combining more than one method, either from data levels or algorithm-level techniques, to enhance prediction [34]. An example of the hybrid approach is ensemble learning. Ensemble learning involves enhancing prediction by using a combination of weak learners to form a strong learner. The major course of error in machine learning is the presence of noise, variance, and bias in the dataset. Ensemble classifiers are built to minimize these factors, which improves the stability and learning performance of machine learning algorithms. A study by Zhou et al. [35] shows a broad overview of why and how ensemble learning improves prediction performance. Diverse ensemble learning strategies that focus on imbalanced learning have been proposed in the literature. For instance, Galar et al. [36] provide a broad overview of different combinations of multiple classifiers to improve predictive accuracy. The ensemble approach can either be constructed using boosting or bagging learning structures to optimize accuracy. The implementation of boosting learning can be found in AdaBoost [37], SMOTEBoost [38]. The bagging implementation that is bootstrap aggregating [39] can be seen in SMOTEBagging [40].

Combining ensemble learning with a data level approach (under-sampling or over-sampling) to solve the imbalanced classification problem has led to several proposals in the literature, with positive results [41]. Although ensemble learning is known to enhance machine learning model performance [42], the arrangement of classifiers alone cannot solve the class imbalance problem. Hence, the ensemble approach needs to be explicitly designed for imbalanced learning to deal with imbalanced classification challenges. For example, The balance-cascading and easy-ensemble algorithms presented in the study by Liu et al. [14,43] use the under-sampling technique with an ensemble approach to train the weak learners and then combine the result to form a robust classifier. These algorithms use the under-sampling method because of its advantage of less training time. They then focus on tackling its disadvantage, which is a reduction of informative samples. Easy-ensemble involves resampling the majority class into several subsets, then training each subset using weak learners (such as AdaBoost [44]) while keeping the minority class constant.

The result of each data subset will then be combined using majority voting. This approach has recoded positive results, which has led to more advances in this direction. An easy-synthetic minority over-sampling technique (easy-SMT) was developed by Wu et al. [4]. Easy-SMT is an integrated ensemble-based method that uses a SMOTE-based oversampling and under-sampling strategy to transfer imbalanced problems into an ensemble-based balance sub-problem. Using an easy-ensemble or balance-cascade algorithm to resample the dataset involves exploring the data samples ignored by the random under-sampling technique. However, both methods keep the minority class constant while training the subsets, creating computational costs if the data is large. Wankhade et al. [45] proposed a hybrid method to deal with an imbalance classification problem that addresses the above challenge. Their technique uses a combination of classification and clustering to enhance recognition of the rare class during learning. Likewise, Vluymnas et al. [46] proposed a hybrid method for solving the imbalanced problem, which combines a preprocessing and classification model. The results of both approaches show an improvement in predicting minority class. Another hybrid approach is developed by Le et al. [47] to predict bankruptcy. Their algorithm uses an over-sampling technique and cost-sensitive learning to handle imbalanced classification problems. The results show that the approach outperforms other existing methods in predicting bankruptcy, which is rare in the dataset used. Application of Imbalance learning has also been seen in rotating machinery; Yuyah et al. [7] show oversampling and future-leaning to handle imbalanced data in fault diagnosis. Different studies have also demonstrated how imbalanced data problems can be handled using deep learning [48,49].

As highlighted above, most of the methods are validated on diverse individual datasets, making them domain-specific. Thus, the peculiarities of our dataset make it challenging to apply off-shelf techniques. Among the different approaches, the hybrid methods show effectiveness and robustness in handling the imbalance problem compared to other single methods. Thus, it motivated this study. Therefore, this study aims to advance the ensemble and hybrid approach by considering the challenge of extremely imbalanced classification problems in the big data domain. Also, our proposed method is inspired by two observations: first, the possibility of convergence of different boosting algorithms for an optimal solution heading to the direction of the gradient of the objective function, and the cost-insensitive predictor can then asymptotically minimize. Second, ensemble algorithms can perform shift decision threshold and calibration of probability estimation, which accounts for class imbalance [50].

5.1.3 Methodology

This section describes the methodology for this study.

5.1.3.1 Derivation of Cost-Sensitive Decision Tree Algorithm

In the proposed approach cost-sensitive decision tree algorithm is used as a weak classifier.

In machine learning, classification involves predicting the class of a given data point, say y_i of a dataset Ds, given their k features $x_i \in R^k$. Classification in predictive modelling is about approximating a mapping function f (·) that minimizes the expected value of some specified loss function $L\left(y_{i,} F(x)\right)$, to make a prediction c_i of the class of each example using its input variables x_i .

$$\widehat{\boldsymbol{F}} = \underset{\boldsymbol{\gamma}}{\boldsymbol{argmax}}\ \boldsymbol{E}_{x,y}[\boldsymbol{L}(\boldsymbol{y}, \boldsymbol{f}(\boldsymbol{x}))] \ , \qquad (5\text{- }1)$$

where γ is the learning rate

Similarly, as described by Hastie et al. [51], the gradient boosting methods uses a real value of $y_i \in R$ and then seek an approximation of $\hat{F}(x)$ that minimize the average value of loss function on the training dataset, this is achieved by starting with a constant function $F_0\,(x)$ and increment it greedily.

$$\boldsymbol{F_0}\,(\boldsymbol{x}) = \underset{\boldsymbol{F}}{\boldsymbol{argmax}}\ \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(\boldsymbol{L}(\boldsymbol{y_i}, \boldsymbol{\gamma})) \qquad (5\text{- }2)$$

$$F_m\,(x) = F(x)_{m-1}\,(x) + \underset{h_m \in H}{argmax}\ [\sum_{i=1}^{n}(L(y_i, F(x)_{m-1} + h_m\,(x_i)\,)] \qquad (5\text{- }3)$$

$h_m \in H$ is the base learner function.

To further minimize the problem, the steepest descent approach is used to transform (eq. 5-1) as the gradient descent and take the derivatives with respect to F_i for $i \in \{1, \dots m\}$

$$\hat{F}_m\,(x) = \hat{F}(x)_{m-1}\,(x) + \gamma_m[\sum_{i=1}^{n} \nabla F(x)_{m-1}\,(L(y_i, F(x)_{m-1} + F(x)_{m-1}\,(x_i)\,)]$$

$$\gamma_m = \underset{\gamma}{argmax}\ [\sum_{i=1}^{n}(L(y_i, F(x)_{m-1} - \nabla F(x)_{m-1}\,L(y_i, F(x)_{m-1}\,)] \qquad (5\text{- }4)$$

To improve the quality of fit of each base learner function, we use the Friedman approach [52],

considering m^{th} steps to fit a decision tree $h_m\,(x_i)$, and j_m are the leaves nodes, we get

$$F_m(x) = F(x)_{m-1} + \sum_{j=1}^{j_m} \gamma_{j_m} 1R_{j_m}(x), \;\; \gamma_{j_m} = \underset{\gamma}{argmax} \sum_{x_{i \in R_{j_m}}}^{n} L(y_i, F(x_i) + \gamma) \qquad (5\text{- }5)$$

j , denotes the number of terminal leave notes in the tree, R is a real values.

Gradient Boosting Tree Algorithm

Input: the training set $\{(\boldsymbol{x_i}, \boldsymbol{y_i})\}_{\boldsymbol{i=1}}^{\boldsymbol{n}}$, a differentiable loss function $L(y_i, F(x_i))$ and number of iterations M.

```
1. Initialize the model with a constant value
        $F_0(x) = \underset{\gamma}{argmax} \sum_{i=1}^{n}(L(y_i, \gamma))$
2. For $m \in \{1, \dots M\}$:
        a. compute $\gamma_m = -[\frac{\delta L(y_i, F(x_i))}{\partial\, F(x_i)}]$; for i= 1,…M
        b. Fit a base learner $h_m$ (using CS-DT) $\partial_m\, to\, (x_{i,\ \gamma_{mi}})$ for i =1,…n
        c. compute multiplier $\gamma_m$ using the following optimization function.
        $F_m(x) = F(x)_{m-1} + \sum_{j=1}^{j_m} \gamma_{j_m} 1R_{j_m}(x), \;\; \gamma_{j_m} = \underset{\gamma}{argmax} \sum_{x_{i \in R_{j_m}}}^{n} L(y_i, F(x_i) + \gamma)$
        d. update the model $F_m(x) = F(x)_{m-1} + \gamma_m \nabla_m(x)$
3. Output $F_M(x)$ = 0
```

Algorithm 1 forms the core component of our proposed approach. It is used as a weak classifier.

5.1.3.2 Our proposed approach

This study aims to enhance the learning algorithm's performance to improve True Positive Rate (TPR) and reduce False Positive Rate (TPR) while learning from the extremely imbalanced system log dataset. Therefore, we combine three machine-learning approaches to form a hybrid algorithm, which enhances the classification of an extremely imbalanced dataset.

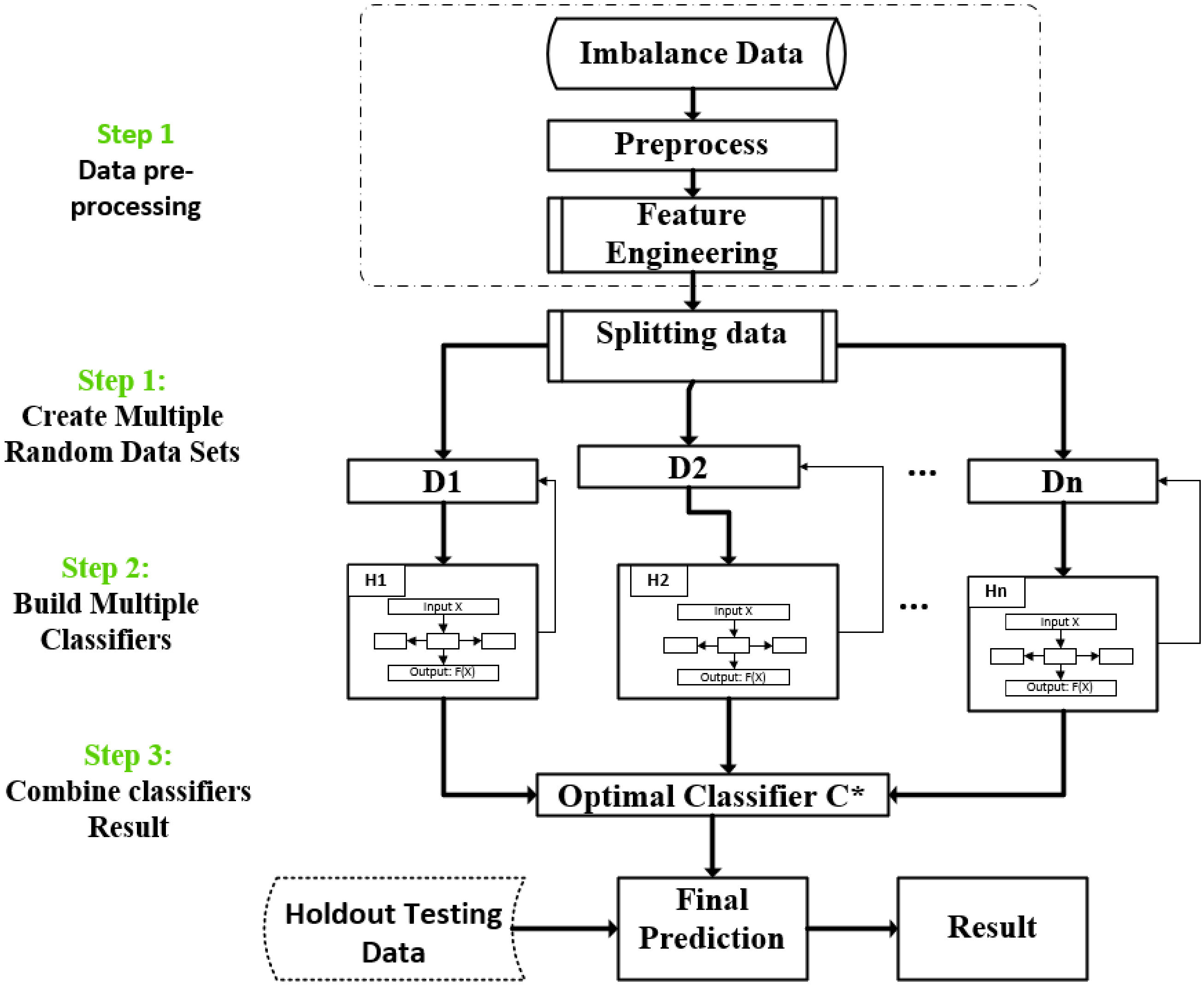


Figure 5- 3 The Methodology of Imbalance learning using BACHE Algorithm

The three approaches are the data level (under-sampling), ensemble learning, and cost-sensitive learning to develop a novel imbalanced learning algorithm called the BACHE algorithm. In the under-sampling phase, a balance-cascading algorithm is used to reduce data from the majority class. The choice of the balance-cascading algorithm is because of its low computation cost and its effectiveness in utilizing the majority class samples ignored by random under-sampling techniques. The ensemble learner approach is chosen because of its strength in combining multiple weak learners to produce a robust classifier. The calibrated cost-sensitive is used to define the cost of misclassification in each weak learner's prediction, which helps in tackling problems where the costs of different types of erroneous predictions are not equal. In the ensemble boosting phase, instead of using a standard decision tree, a cost of classification using a calibrated probability estimate is considered at each iteration by modifying the updating rule with regards to the modified loss function. Likewise, instead of finding the best classifier, the problem is directed to focus on finding the best learning rate γ [53–55].

Therefore, what makes a difference here is the tree structure and the model weight updating rule. The BACHE algorithm works as follows; first, data preprocessing and feature engineering is conducted. After preprocessing the dataset and selecting the right features, the data is divided into two. 80% of the data is kept for model training and 20% for model testing. Then the dataset is divided into several subsets using a cascading balanced approach [56]. At every boosting integration step (selection with replacement), the samples of each subset are balanced to form Balanced Data ($D_{i's}$). After the dividing and balancing process, each subset is trained using weak learners. The process continues for the number of defined iterations. At each iteration, the subset learns using a weak learner (H_{i^s}) at the end of the ensemble process, the result of all the weak learners, is combined to get a hybrid ensemble classifier. The final model is then evaluated using new hold-out datasets.

As seen in Figure 5-3, the proposed BACHE methodology explores both the majority class ($\boldsymbol{N}$) and Minority class ($\boldsymbol{P}$) in a supervised learning manner. The weak learners H_{i^s} are trained in sequence on a weighted version of the dataset using a cost-sensitive boosting algorithm. Considering N in the under-sampling process, if data point example say $\boldsymbol{x_i} \in \boldsymbol{N}$ is correctly classified to be in $\boldsymbol{N}$ it easier to infer that $\boldsymbol{x_i}$ is reasonably redundant in N, given that we already have the outcome as H_1 [57]. Therefore, x_i will be removed from N. (That shows N will be reduced after training each H_i). Every H_i deals with balanced sub-set $|N_i| = |P_i|$, after processing all the subsets of the cascaded dataset, the outcome of H_{i^s} is combined using a weighted majority vote.

Elaborately, considering the majority class N and minority class P, the length of iteration is S_i and the length of each $n \in N$ subset is defined as M (we use an under-sampling technique to split N into random subsets $n_{1,} n_2, n_3 \ldots n_T \in N$). Then a subset of $p \in P$ is combined with each $n \in N$ to form a balanced sub-dataset (D_i). These D_{i^s} are trained using weak classifiers, which are later combined to form an optimized classifier. In each weak classifier, a cost-sensitive calibrating boosting algorithm is used. Such as adaMEC [50], a score of the form $(x) \in [P, N]$ is generated.

A probability of x belonging to a positive class P is given as $prb(y = (1|x))$, x will be assigned to a class with a minimized expected cost. In other words, a data point x_i will be assign to positive class P if and only if $prb\big(y = (1|x)\big)cv > \; prb\big(y = (-1|x)\big) \leftrightarrow prb\big(y = (1|x)\big) > \frac{1}{1+c}$. For example, using the imbalance learning cost matrix (see Table 5-1)

$$c = \begin{bmatrix} 0 & 1 \\ c & 0 \end{bmatrix}, \text{c } (y_i) = \begin{cases} c \;\; if \; y_i \; = 1 \\ 0 \;\; if \; y_i \; = -1 \end{cases} \tag{5- 6}$$

Where $prb\big(y = (-1|x)\big) = 1 - prb\big(y = (1|x)\big)$.

Otherwise data point x_i is assigned to the negative class N. It is important to note that probability estimates are not always straightforward to obtain from a classifier's outputs [58]. Therefore, a generated score of the form (x) is calibrated using platt scaling. The classification of the extreme minority is accounted for in the calibration step as detailed in [58]. In the Platt calibration, it uses $\frac{P+1}{P+2}$ for positive class and $\frac{1}{N+2}$ for negative class, rather than 1 and 0 as the target probability estimation of the P and N. Therefore, in BACHE we aim to reduce the ensemble error rate by focusing on different positive class P, as we want to model P better to enhance detection of the extreme minority and also avoiding accuracy degradation for the negative class N. The BACHE algorithm pseudocode is presented in algorithm 2.

```
The  Balanced Calibrated Hybrid Ensemble Technique (BACHE) Algorithm
INPUT:
            Dataset: $\mathbf{D} = \{(\boldsymbol{x_i}, \boldsymbol{y_i})\}_{\boldsymbol{i=1}}^{\boldsymbol{n}}$ with minority class $\boldsymbol{P}$, majority class $\boldsymbol{N}$
             and $\boldsymbol{P} < \boldsymbol{N}$ .
            The number iteration or the number of subsets to be sampled from
             N: $\boldsymbol{M}$ ,
             Imbalance Ratio $\boldsymbol{IR} = \frac{\boldsymbol{P}}{\boldsymbol{N}} * 100$
            The number of iterations to train the calibrating ensemble $\boldsymbol{H_i}: \boldsymbol{s_i}$
$\boldsymbol{for\ i = 1\ to\ K\ Do:}$
    $\boldsymbol{f} \leftarrow \sqrt[\boldsymbol{M-1}]{\frac{\boldsymbol{n+}}{\boldsymbol{n-}}}$ , $\boldsymbol{f}$ is the FP-rate that $\boldsymbol{H_i}$ should achieve.
    Randomly sample a subset $\boldsymbol{N_i}$ of $\boldsymbol{n_+}$ with replacement.
            $\boldsymbol{N'} = (\boldsymbol{IR} > \mathbf{1.25}\ (\boldsymbol{M_2} - \mathbf{1}),\ \boldsymbol{K})$;  $\boldsymbol{P} = \boldsymbol{P} + \boldsymbol{P'}$
    $\boldsymbol{for\ i = 1\ to\ j\ Do:}$
        Training Phase:
         Split the data in training set $D_t$ and calibration set $D_c$ (for correcting distortion)
        On $\boldsymbol{D_t}$:
            Train $\boldsymbol{H_i}$ using $\boldsymbol{P'}$ U $\boldsymbol{N_i}$ .
             $\boldsymbol{H_i}$ is obtained using Algorithm 1 with $\boldsymbol{S_i}$ as weak classifier
             $\boldsymbol{h_{i,j}}$ and corresponding weight $\boldsymbol{\alpha_{i,j}}$ .
            The ensemble shifted decision threshold is $\boldsymbol{\theta_i}$
        On $\boldsymbol{D_c}$- calibrated boosting:
            a. calculate score $s(x_i)$= $\frac{\sum_{\tau:} h_{i,j}(x_i)=1}{\sum_{\tau=1}^{s_i} a_{i,j}} \in [1,0] \forall x_i \in D_c$
            b. calculate number of P and N in $D_c$;
                find A,B s.t $\sum_{i \in D_c} prb(y = (1|x_i) - y_i)^2$ is minimized.
                Where $prb(y = (1|x) = \frac{1}{1+e^{As(x)}+B}$ and $y_i = \begin{cases} \frac{P+1}{P+2} & if\ y_{i=positive} \\ \frac{1}{N+1} & if\ y_{i=negative} \end{cases}$
        Prediction Phase:
            On new data-point $(x)$:
                Calculate prior weight score $s(x)$
                Obtain prior weight probability estimate $prb_w\ (y = (1|x) = \frac{1}{1+e^{As(x)}+B})$
                Predict class H($x_j$) = $\overset{sign}{\gamma}\ [prb_w\ (y = (1|x) > \boldsymbol{\theta_j}]$
                Adjust $\boldsymbol{\theta_j}$ such that $\boldsymbol{H_{ij^s}}$ the false positive rate is $\boldsymbol{f}$
                Remove from $\boldsymbol{N}$ all samples that are correctly classified by $\boldsymbol{H_j}$ .
OUTPUT ENSEMBLE:
    Return  $\boldsymbol{H_i(x)} = \overset{\boldsymbol{sign}}{\boldsymbol{\gamma}}\ (\sum_{\boldsymbol{i=1}}^{\boldsymbol{M}} \sum_{\boldsymbol{j=1}}^{\boldsymbol{S_i}} \boldsymbol{\alpha}_{\boldsymbol{i,j}}\ \boldsymbol{h_{i,j}}\ (\boldsymbol{x}) - \sum_{\boldsymbol{i=1}}^{\boldsymbol{M}} \boldsymbol{\theta}_{\boldsymbol{i}})$
```

5.1.3.4 Experiment

To validate the effectiveness of the proposed approach, we use the following datasets as input. The first data is the data generated from the central maintenance system (log-based CMS data), and the second data is the record of maintenance activities. The datasets are obtained from a fleet of long-range (A330) aircraft and A320 families. According to families, aircraft grouping is necessary because the data generated differ in properties and structure. The designation routes were different for each family; some were mainly used for long-distance routes, while some were primarily used for short distances. From the A330 aircraft family, the total number of failure/warning messages after preprocessing is 389902, and the A320 family has a total of 890120.

The main objective is to develop a predictive model to predict failure resulting in aircraft's unplanned repairs or components' replacement. Therefore, we choose target components identified by Functional Item Number (FIN). The representation of these components is extremely rare. The basic idea is to correctly detect the extreme minority class samples and the majority class samples during model classification.

Apart from the high skewness in the dataset, the raw data has many challenges that require preprocessing, such as data incompleteness, lack of behaviours and trends, containing null values, lacking the features of interest, and containing noise. Therefore, we follow the data knowledge discovery approach [62]. We preprocessed the data and transformed it into a suitable format for machine learning. After that, we carry out the Feature Engineering (FE) process. FE is the integral and critical step of the machine learning process because the model's performance output depends on the quality of data and the right features selected (see chapter 2). After the preprocessing and feature engineering phase. The data is divided into two: For training and for testing the model. The data was split into training and testing divided into 70/30 (from January 2011 to September 2016) and validation data from October 2016 to April 2018 (without known label).

The following requirements are considered in the design and develop the imbalance-learning framework.

1. Features obtained from the raw CMS dataset should adequately represent the component replaced.

2. The baseline learning algorithm and classifier should be suitable for large imbalanced datasets.

3. The model performance evaluation metrics should be suitable for an imbalance scenario.

4. Prognostic alert requirements: - Predictive model should flag up alerts for maintenance needs (component replacement), not more than ten and not less than two flight cycles before failure point. The window period is to avoid early replacement of a component, which will mean underutilizing resources and not too close to failure to give adequate room to prepare for maintenance.

5. Model should achieve more than 60% precision, recall more than 50%, or G-mean of greater than 50%.

From the dataset, we selected a few aircraft components for validation. The Electronic control unit/ Electronic engine unit (4000KS), High-pressure bleed valve (4000HA), pressure regulating valve (4001HA), Satellite data unit (5RV1), Flow control valve (11HB), Avionics equipment ventilation computer(10HQ), Air traffic service unit (1TX1) and Flow control valve (8HB). The selection is based on descriptive analysis, which shows the percentage of each component replaced over the period under consideration. We select the components with the highest number of replacements. We clustered the dataset (failure/ warning message) according to every specific component under consideration, then in each of the clusters, the targeted unplanned component replacements are labelled as a positive class (representing the minority class-P). Simultaneously, all the failure/warning messages before each replacement are labelled as the negative class (representing the majority class-N). In each cluster, since the data is sequential in terms of date-time and flight circles, we group the data into windows using date-time and flight cycles; a window size of 30 aircraft flight cycles was used. The choice of window size is based on the domain of expert advice.

Our experiment aims to compare the performance of state-of-the-art ensemble boosting methods for imbalanced learning with our proposed approach. Therefore, the following experiment was set up to evaluate the proposed BACHE algorithm's performance on aircraft rare unplanned failure prediction problems.

Balance Bagging (BB): This is an ensemble learning method. It uses a bagging approach with an additional capability to balance the training dataset at the fitting time. During training, the parameter can be turned for the best results. Therefore, BB is considered our baseline method since our algorithm is based on the ensemble learning approach and focuses on tackling extremely rare failure problems in aircraft systems. The hyper-parameters are Base_estimator=None,n_estimators=10,max_samples=1.0,max_features=1.0,bootstrap=True,bootstrap_features=False,oob_score=False,warm_start=False,n_jobs=None,random_state=None,verbose=0.

SMOTE-Random Forest (SMT-RF): This method combines an imbalance learning algorithm with an ensemble algorithm. The minority class is first resampled using the SMOTE technique, and then the ensemble-RF approach is used as the classifier. The Random Forest algorithm is implemented using the following hyperparameters: learning_rate = 0.1, Max_depth =10, Subsample = 50, Colsample = 0.3, n_estimator= 10

XGBoost (eXtreme Gradient Boosting): XGBoost is an ensembled learning based algorithm, ensembled are contructed from decision trees, trees are added using boosting approach (one at a time to the enseble as fit for classification) [63]. XGBoost Scikit_Learn API was used with the following hyperparameters**:** learning_rate = 0.1, Max_depth =10, Subsample = 50, Colsample = 0.3, n_estimator= 10.

Cost-Sensitive C4.5 Algorithm: This ensemble-based algorithm builds decision trees from a set of training data [64]; the trees are used for classification. C4.5 algorithm is implemented using the following hyperparameters: learning_rate = 0.1, Max_depth =10, Subsample = 50, Colsample = 0.3, n_estimator= 10.

Balance calibrated Hybrid Ensemble Technique (BACHE): The proposed approach.

Experiment Running Environments:
Programing langege : Python
Machine learning Editor: Sublime and Jupyter notebook
Major Packages: Pandas, Scikit-learn, Keras, TensorFlow, sciPy and more.
Running the experiments with multiple seeds will ensure the approach is not sensitive to different start conditions. Some of the sensitivity to initial conditions could be that the failure distribution can substantially differ between the training and validation datasets,

which will likely negatively affect model training. To mitigate that, stratified samples and random seed can be used so that the proportions of the dependent variable are similar in training, testing and validation dataset. In the implementation, each algorithm was run five times with the same hyperparameter for each target event using five random seeds then the average is obtained.

5.1.4 Results and Discussion

This experiment investigates the proposed approach's performance against the existing ensemble learning algorithms (Balance Bagging as baseline) and hybrid imbalance learning algorithms (SMOTE + Random Forest). The choice of the baseline algorithms is to enable us to assess the proposed method's performance, which uses a cost-sensitive decision tree as a weak classifier and then employs an ensemble approach to get a hybrid algorithm (BACHE) as a solution to the extremely imbalanced classification problem.

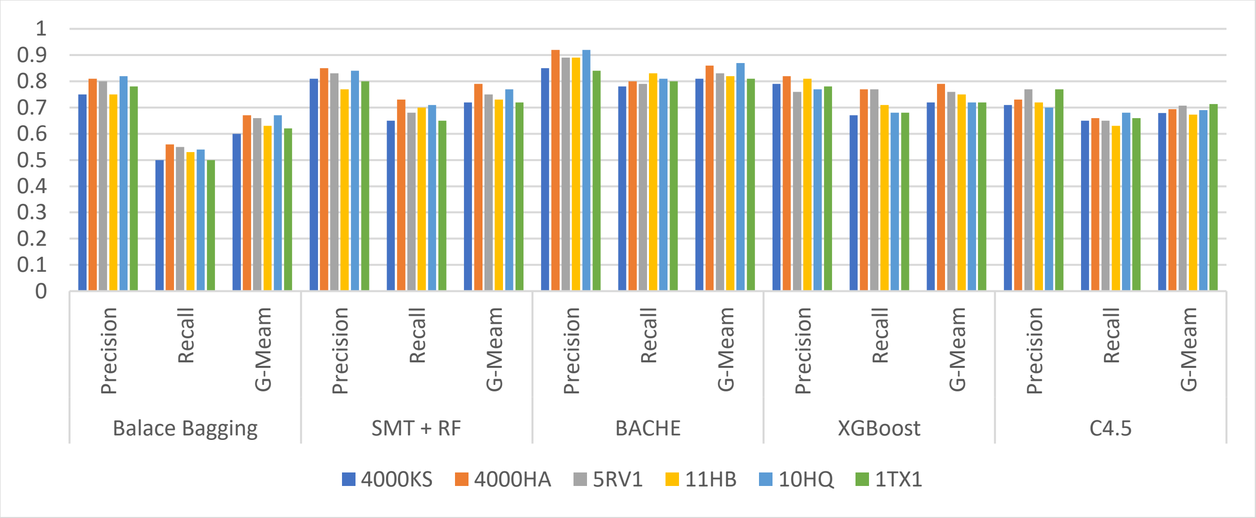


Figure 5- 4 Comparing BACHE against other Algorithms

Table 5- 2 Experimental Result using data from a fleet of A330 and A320 Aircraft family

Dataset (TFIN)	IR%	Balance Bagging (baseline)			SMT +RF			BACHE		
		Precision	Recall	G-mean	Precision	Recall	G-mean	Precision	Recall	G-mean
A330 (Long Range) Family										
4000KS	0.0043	0.75	0.50	0.60	0.81	0.65	0.72	0.85	0.78	0.81
4000HA	0.0047	0.81	0.56	0.67	0.85	0.73	0.79	0.92	0.80	0.86
5RV1	0.0044	0.80	0.55	0.66	0.83	0.68	0.75	0.89	0.79	0.83
A320 (Short Aisle) Family										
11HB	0.0028	0.75	0.53	0.63	0.77	0.70	0.73	0.89	0.83	0.82
10HQ	0.0031	0.82	0.54	0.67	0.84	0.71	0.77	0.92	0.81	0.87
1TX1	0.0021	0.78	0.50	0.62	0.80	0.65	0.72	0.84	0.80	0.81

*TFIN:-Target Functional Item Number, IR:- Imbalance Ratio, SMT:- SMOTE, RF:- Random Forest, BACHE:- Balanced Calibrated Hybrid Ensemble Technique.

Tables 5-2 and Figure 5-5 present the experiment results. It can be observed that in all cases, the proposed BACHE algorithm outperforms the two algorithms in terms of recall and G-mean. The G-man's superior performance indicates the trade-off between recognition in both classes, which is also a good classification effect for imbalanced datasets. Similarly, the high precision suggests that the false positive rate is low, and the high recall score indicates that the BACHE algorithm is sensitive to the minority class. Furthermore, Figure 5-6 shows how BACHE records a significant percentage reduction in false positives compared to other methods. Although, the positive class (the minority class) is extremely rare. However, the BACHE algorithm is robust to skewed distribution by achieving a better result.

It is also important to note that our goal is to achieve a G-mean score of greater than 50% as part of the target requirement for this, which is the mean average of detecting extremely rare failure from the log-based dataset. The higher G-mean score for the BACHE algorithm shows that the model can distinguish the failure patterns leading to unexpected component replacement.

We also evaluate the proposed method's effectiveness in terms of false-positive and true-positive rates, considering the different aircraft families' datasets.

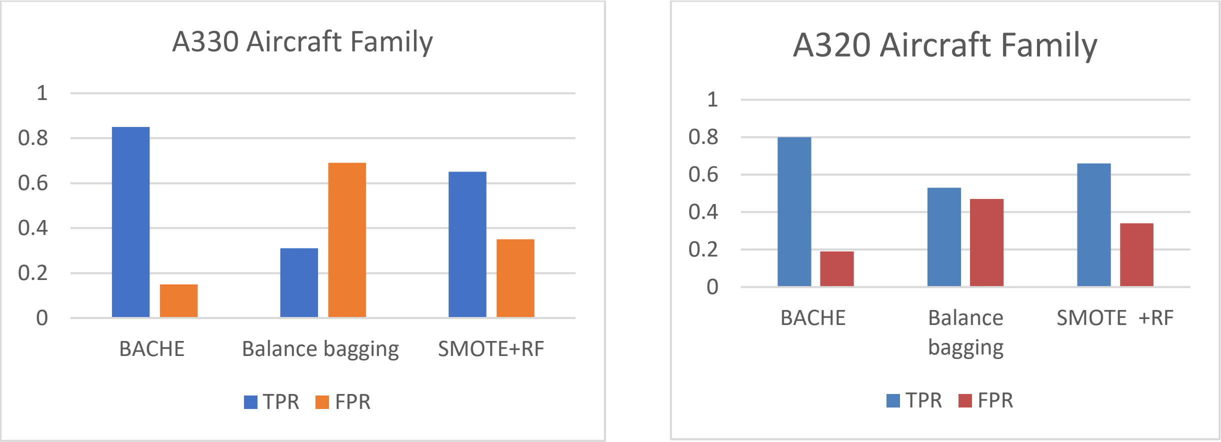
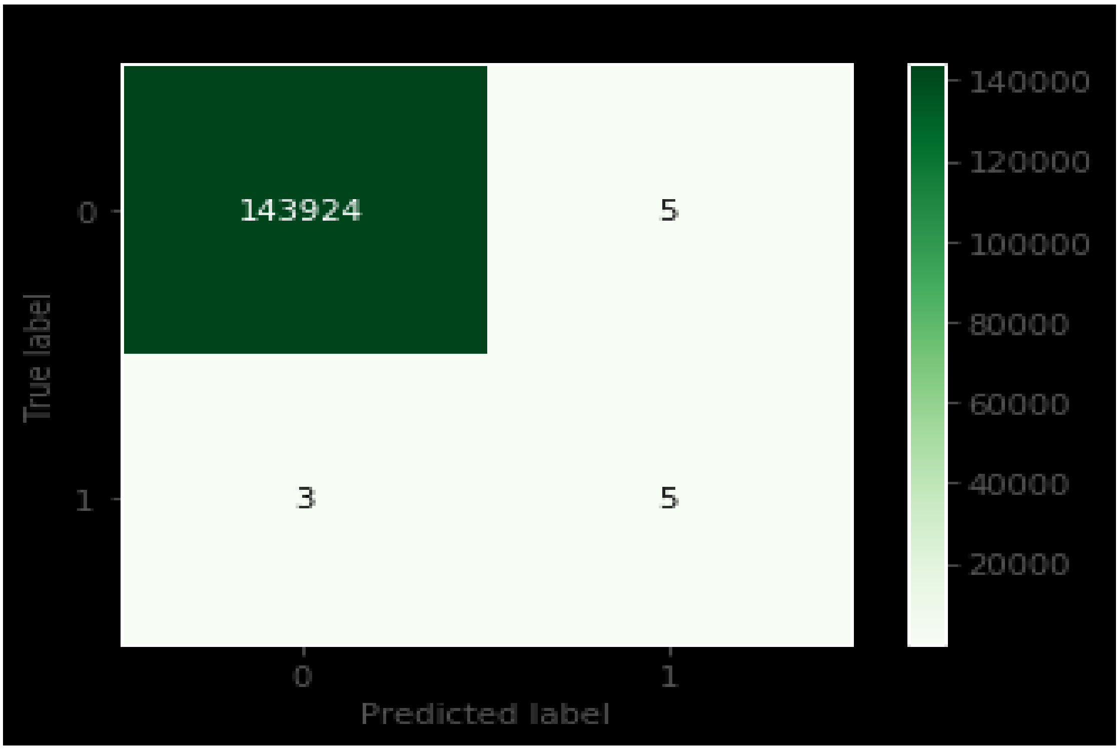


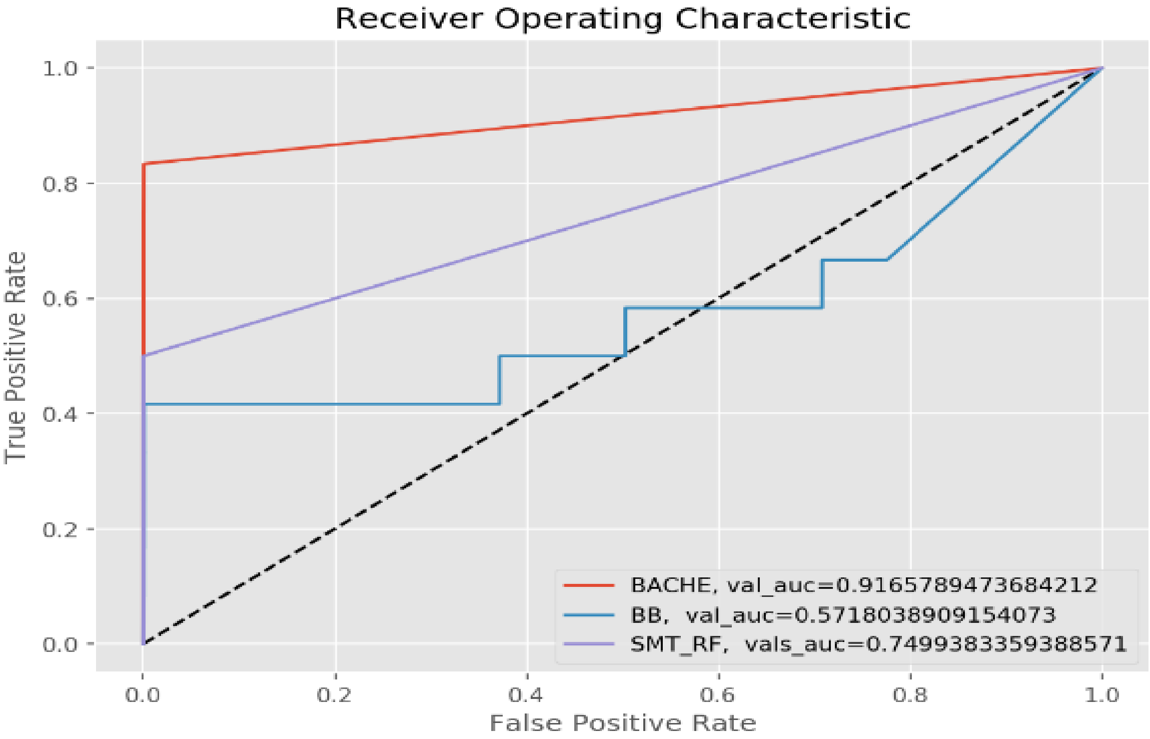
Figure 5- 5 The average overall performance of each algorithm on the two aircraft families (A330 and A320)

Figure 5-5 shows the average FPR and TPR for each algorithm in both A330 and A320 aircraft families. BACHE averagely achieved a better (low) false-positive rate compared to the closest SMOTE+RF. In the A330 family, a balanced bagging algorithm has a predictive performance in terms of FPR of 69%, SMOTE+RF has 35% and BACHE 15%. Comparing BACHE with closes SMOTE+RF, it is clear to see that there is a significant improvement of about 20%. Similarly, in the A320 family, the FPR for balance bagging is 47%, SMOTE+RF is 34%, and BACHE is 19%, showing an improvement of about 15%. The result validates the superior performance of BACHE in different aircraft families in the fleet.

Furthermore, another evaluation is the ROC curve reading, which shows that even though there is a significant percentage of false-positive rate (approximately 15%), the absolute probability is reasonably small.


143924
5
3
5
0
1
True label
0
1
Predicted label
140000
120000
100000
80000
60000
40000
20000

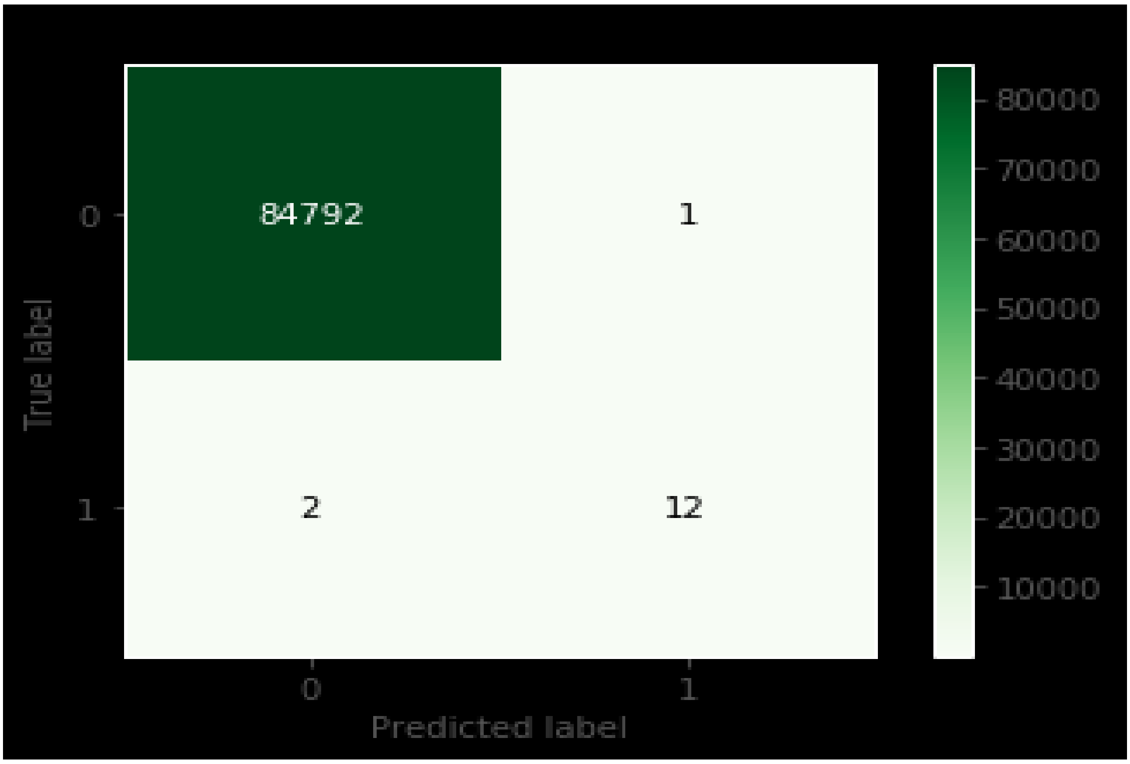
(a)


Receiver Operating Characteristic
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
BACHE, val_auc=0.9165789473684212
BB, val_auc=0.5718038909154073
SMT_RF, vals_auc=0.7499383359388571

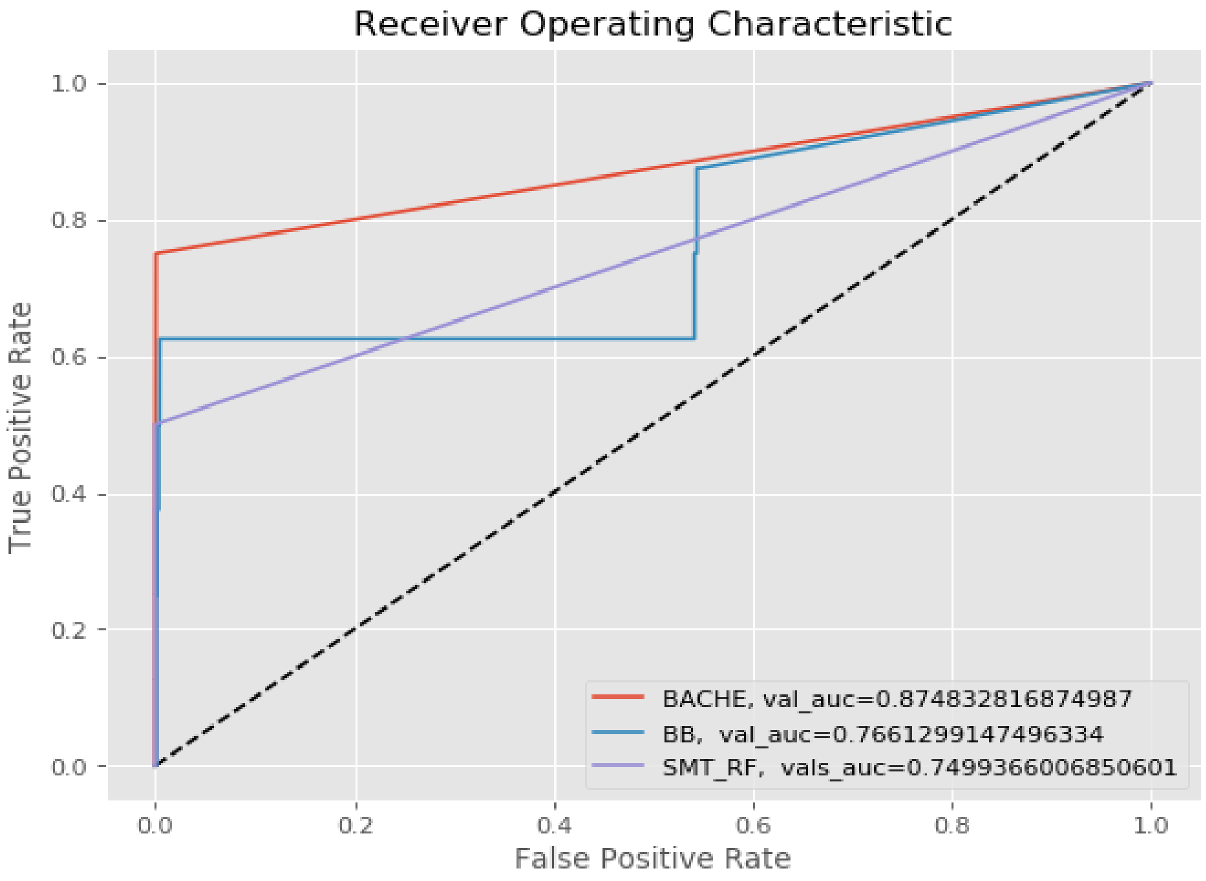
(b)

Figure 5- 6 (a) is the Confusion Matric of BACHE prediction, and (b) the ROC-Curve shows the performance of the three algorithms considered in this study using data from the A330 aircraft family

Figure 5-7 shows that the BACHE algorithm predicted 5 out of 8 unplanned failures, leading to the aircraft's pressure regulating valve replacement (FIN_4000HA). This prediction includes 10 flight cycles in advance. It can be observed that the model detected and predicts approximately 70% of extreme failure, which is a reasonable specificity, especially for aircraft maintenance. The area under the curve is 0.91. This shows that the BACHE algorithm can predict more than 90% of the probability of an observation belonging to each class in the A330 aircraft family.



(a)



(b)

Figure 5- 7 (a) is the Confusion Matric of BACHE prediction, and (b) ROC-Curve shows the performance of the three algorithms considered in this study using data from the A320 aircraft family

Figure 5-8 shows the predictive performance of BACHE on the A320 aircraft family. The result indicates that the BACHE algorithm predicted 12 out of 14 unplanned failures, leading to the aircraft flow control valve (FIN_11HB). The area under the curve is 0.87. The BACHE algorithm can predict more than 85% probabilities of an observation belonging to each class in the A320 aircraft family.

We presented a confusion matrix and ROC for target functional items 4000HA and 11HB because the prediction performance is at the same range for other components in each aircraft family. We considered the remaining components from the A330 family, the electronic control unit/ electronic engine unit (4000KS), the satellite data unit (5RV1). The A320 family are the avionics equipment ventilation computer (10HQ), and the air traffic service unit (1TX1).

Also, it can be observed that the imbalanced ratio has an impact on performance. For instance, looking at Table 5-2 in cases where the IR is low, we obtain a lower G-mean compared to the ones with higher IR. For instance, in the A320 family, 1TX1 has the lowest IR of 0.21% and a G-mean score of 0.81, Compared to 10HQ with the highest IR of 0.31% and G-mean score of 0.87. Similar performance can be seen in the A330 family, where 4000KS has the lowest IR of 0.43% and the G-mean score is 0.81 compared to 400HA with the highest IR of 0.47% and G-mean score of 0.86. Despite the extremely imbalanced ratio in all the cases considered, our proposed algorithm still achieved better performance compared to other similar algorithms.

Another data factor that can impact the algorithm is the class small disjunct. Small disjunct arises when data in the same class is represented with different clusters (within class imbalance). The less represented small sub-clusters can further worsen classification performance degradation in an extreme imbalance dataset. We handled the challenge of class small disjunct problems intrinsically in the BACHE algorithm by clustering each class independently to identify clusters in each class. We subsequently oversampled sub-clusters in each class so that clusters in each class are balanced before the classification step.

One of the objectives of this study is the performance optimization of an imbalance learning algorithm. Evolution of the proposed BACHE against other similar algorithm was performed, the result displayed in Figure 5-9. Running each algorithm for classification of individual component failure. The result indicates that balance bagging has the fastest training time (averagely 20 seconds), with the XGBoost algorithm having the worst training time (averagely 60 seconds). In contrast to the Proposed BACHE algorithm, which has an average training time of 50 seconds. Although Balance bagging and Random Forest (RF) show less computation time than BACHE, as observed, the difference is less than 20 seconds for balanced bagging and less than 10 seconds for the random forest. On the other hand, BACHE performed better in precision, recall and G-mean (see table 2). Mis-classifying an example from the majority class as an example from the minority class is called a false-positive. False-positive is often not desired but less critical than classifying an instance from the minority class as belonging to the majority class, known as a false-negative. In the context of this study, false-negative means misclassifying fault as healthy, very critical as it can lead to equipment damage. In this study, false-positive means misclassifying a healthy component as a faulty component. This can result in the extra cost of maintenance checking. BACHE high precision indicates a less number of False Positives, and high recall means fewer False Negatives.

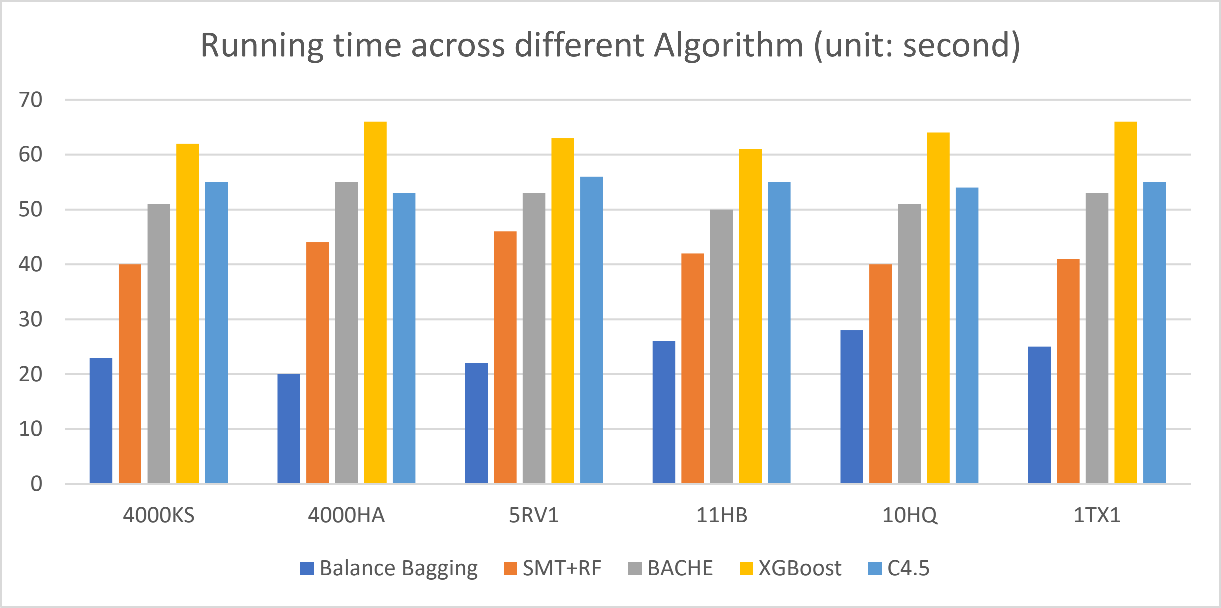


Figure 5- 8 Comparing running time between the BACHE algorithm against other ensemble-based algorithms

G-mean is a metric that measures the balance between classification performances on both the majority and minority classes. G-mean measures the root of the product of class-wise sensitivity; it attempts to maximise each class's accuracy and keeps the accuracy balanced. It is a performance metric that correlates both. A low G-Mean indicates poor performance in the classification of the positive cases even if the negative cases are correctly classified as such. This measure is important in the avoidance of overfitting the negative class and underfitting the positive class. The algorithm can classify samples from both minority and majority classes which is shown in higher G-mean for BACHE compared to others.

5.1.5 Conclusion

In this paper, a novel imbalance-learning algorithm is proposed and developed; we develop a model for predicting aircraft component replacement using real-world test cases from the log-based central maintenance system data. The new imbalance algorithm is based on the Balance-Calibrated Hybrid Ensemble Technique

(BACHE). It is designed to handle extremely imbalanced classification problems. It focuses on improving the detection of a rare failure in the aircraft maintenance predictive models. The experiment showed that the BACHE algorithm has a better performance than other similar ensemble and imbalanced learning techniques. The novel approach also achieved a significant level of improvement in the reduction of false-positive and false-negative rates, which is one of the targets of this study. The results showed that the model could predict aircraft component replacement within the defined range; this contribution can enhance predictive maintenance in fleet reliability analysis. The model, when validated, can be used for predictive aircraft maintenance to improve the efficiency of the component replacement prognostic model. In the future, we hope to develop this work further by looking at the effect of class overlapping in the process of over-sampling the minority class in the imbalanced learning context. We also hope to explore the applicability in an online heterogeneous dataset.

5.1.6 References

1. Eickmeyer J., Li P., Givehchi O., Pethig F., Niggemann O. Data Driven Modeling for System-Level Condition Monitoring on Wind Power Plants. Int. Work. Princ. Diagnosis. 2015; 1507: 43–50.

2. Sahal R., Breslin JG., Ali MI. Big data and stream processing platforms for Industry 4.0 requirements mapping for a predictive maintenance use case. Journal of Manufacturing Systems. Elsevier; 2020; 54(November 2019): 138–151. Available at: DOI:10.1016/j.jmsy.2019.11.004

3. Dangut MD., Skaf Z., Jennions IK. An integrated machine learning model for aircraft components rare failure prognostics with log-based dataset. ISA Transactions. Elsevier Ltd; 2020; (xxxx). Available at: DOI:10.1016/j.isatra.2020.05.001

4. Wu Z., Lin W., Ji Y. An Integrated Ensemble Learning Model for Imbalanced Fault Diagnostics and Prognostics. IEEE Access. 2018; 6: 8394–8402. Available at: DOI:10.1109/ACCESS.2018.2807121

5. Wang J., Ma Y., Zhang L., Gao RX., Wu D. Deep learning for smart manufacturing: Methods and applications. J. Manuf. Syst. The Society of Manufacturing Engineers; 2018; : 1–13. Available at: DOI:10.1016/j.jmsy.2018.01.003

6. He H. Imbalanced Learning. Self-Adaptive Systems for Machine Intelligence. New Jersey: John Wiley & Sons, Inc.,Hoboken, New Jersey.; 2011. 44–107 p. Available at: DOI:10.1002/9781118025604.ch3

7. Zhang Y., Li X., Gao L., Wang L., Wen L. Imbalanced data fault diagnosis of rotating machinery using synthetic oversampling and feature learning. Journal of Manufacturing Systems. Elsevier; 2018; 48(August 2017): 34–50. Available at: DOI:10.1016/j.jmsy.2018.04.005

8. Lee DH., Yang JK., Lee CH., Kim KJ. A data-driven approach to selection of critical process steps in the semiconductor manufacturing process considering missing and imbalanced data. Journal of Manufacturing Systems. Elsevier; 2019; 52(May): 146–156. Available at: DOI:10.1016/j.jmsy.2019.07.001

9. Tao F., Qi Q., Liu A., Kusiak A. Data-driven smart manufacturing. Journal of Manufacturing Systems. The Society of Manufacturing Engineers; 2018; 48: 157–169. Available at: DOI:10.1016/j.jmsy.2018.01.006

10. Branco P., Torgo L., Ribeiro RP. A Survey of Predictive Modeling on Imbalanced Domains. ACM Computing Surveys. 2016; 49(2): 1–50. Available at: DOI:10.1145/2907070

11. Nghiem LT., Thu TT., Nghiem TT. MASI: Moving to adaptive samples in imbalanced credit card dataset for classification. 2018 IEEE International Conference on Innovative Research and Development, ICIRD 2018. 2018. pp. 1–5. Available at: DOI:10.1109/ICIRD.2018.8376315

12. Sajana T., Narasingarao MR. A comparative study on imbalanced malaria disease diagnosis using machine learning techniques. Journal of Advanced Research in Dynamical and Control Systems. 2018; 10: 552–561. Available at: DOI:https://www.jardcs.org/backissues/abstract.php?archiveid=2962&action=fulltext&uri=/backissues/abstract.php?archiveid=2962

13. Jiao Z., Jia G., Cai Y. A new approach to oil spill detection that combines deep learning with unmanned aerial vehicles. Computers and Industrial Engineering. Elsevier; 2018; (September): 1–12. Available at: DOI:10.1016/j.cie.2018.11.008

14. Liu XY., Wu J., Zhou ZH. Exploratory under-sampling for class-imbalance learning. Proceedings - IEEE International Conference on Data Mining, ICDM. IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS; 2006. pp. 965–969. Available at: DOI:10.1109/ICDM.2006.68 (Accessed: 27 January 2019)

15. Lu Y., Cheung Y-M., Tang YY. Bayes Imbalance Impact Index: A Measure of Class Imbalanced Data Set for Classification Problem. IEEE Transactions on Neural Networks and Learning Systems. 2019; 1(c): 1–15. Available at: DOI:10.1109/tnnls.2019.2944962

16. Ali A., Shamsuddin SM., Ralescu AL. Classification with class imbalance problem: A review. International Journal of Advances in Soft Computing and its Applications. 2015; 7(3): 176–204.

17. Chang F., Zhou G., Zhang C., Xiao Z., Wang C. A service-oriented dynamic multi-

level maintenance grouping strategy based on prediction information of multi-component systems. Journal of Manufacturing Systems. Elsevier; 2019; 53(October 2018): 49–61. Available at: DOI:10.1016/j.jmsy.2019.09.005

18. Ning F., Shi Y., Cai M., Xu W., Zhang X. Manufacturing cost estimation based on a deep-learning method. Journal of Manufacturing Systems. Elsevier; 2020; 54(December 2019): 186–195. Available at: DOI:10.1016/j.jmsy.2019.12.005

19. Fernández Alberto, Garcia Salvador, Galar Mikel, Prati Ronaldo, Krawczyk Bartosz HF. Learning From Imbalanced Data Sets. 2018. Available at: DOI:https://link.springer.com/content/pdf/10.1007%2F978-3-319-98074-4.pdf (Accessed: 6 May 2019)

20. Haixiang G., Yijing L., Shang J., Mingyun G., Yuanyue H., Bing G. Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications. 2017; 73: 220–239. Available at: DOI:10.1016/j.eswa.2016.12.035

21. Abd Elrahman SM., Abraham A. A Review of Class Imbalance Problem. Journal of Network and Innovative Computing. 2013. Available at: DOI:www.mirlabs.net/jnic/index.html (Accessed: 23 January 2019)

22. Chawla N V., Lazarevic A., Hall LO., Bowyer KW. SMOTEBoost : Improving Prediction. Lavrač N., Gamberger D., Todorovski L., Blockeel H. (eds) Knowledge Discovery in Databases: PKDD 2003. LNCS. 2003. pp. 107–119. Available at: https://link.springer.com/content/pdf/10.1007%2F978-3-540-39804-2_12.pdf (Accessed: 7 February 2019)

23. Wu Z., Lin W., Ji Y. An Integrated Ensemble Learning Model for Imbalanced Fault Diagnostics and Prognostics. IEEE Access. 2018; 6: 8394–8402. Available at: DOI:10.1109/ACCESS.2018.2807121

24. Chawla N V., Lazarevic A., Hall LO., Bowyer KW. SMOTEBoost: Improving Prediction of the Minority Class in Boosting. 2003; : 107–119. Available at: DOI:10.1007/978-3-540-39804-2_12

25. Chawla N V., Lazarevic A., Hall LO., Bowyer KW. SMOTEBoost: Improving Prediction of the Minority Class in Boosting. 2003; : 107–119. Available at: DOI:10.1007/978-3-540-39804-2_12

26. Sun M., Qian H., Zhu K., Guan D., Wang R. Ensemble learning and SMOTE based fault diagnosis system in self-organizing cellular networks. 2017 IEEE Global Communications Conference, GLOBECOM 2017 - Proceedings. 2018. pp. 1–6. Available at: DOI:10.1109/GLOCOM.2017.8254569

27. Han H., Wang WY., Mao BH. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. Lecture Notes in Computer Science. 2005. pp. 878–887. Available at: DOI:10.1007/11538059_91

28. Domingos P., Ling CX., Sheng VS. MetaCost-AGeneralMethodforMakingClassifiersCostSensitivity. Encyclopedia of Machine Learning. 2008; : 231–235. Available at: DOI:10.1.1.15.7095

29. Bahnsen AC., Aouada D., Ottersten B. Example-dependent cost-sensitive decision trees. Expert Systems with Applications. 2015; 42(19): 6609–6619. Available at: DOI:10.1016/j.eswa.2015.04.042

30. Lu H., Xu Y., Ye M., Yan K., Gao Z., Jin Q. Learning misclassification costs for imbalanced classification on gene expression data. BMC Bioinformatics. BMC Bioinformatics; 2019; 20(Suppl 25): 1–10. Available at: DOI:10.1186/s12859-019-3255-x

31. Maheshwari S., Jain RC., Jadon RS. An insight into rare class problem: Analysis and potential solutions. Journal of Computer Science. 2018; 14(6): 777–792. Available at: DOI:10.3844/jcssp.2018.777.792

32. Liu XY., Zhou ZH. The influence of class imbalance on cost-sensitive learning: An empirical study. Proceedings - IEEE International Conference on Data Mining, ICDM. 2006; : 970–974. Available at: DOI:10.1109/ICDM.2006.158

33. Zhao P., Zhang Y., Wu M., Hoi SCH., Tan M., Huang J. Adaptive Cost-Sensitive Online Classification. IEEE Transactions on Knowledge and Data Engineering. 2019; 31(2): 214–228. Available at: DOI:10.1109/TKDE.2018.2826011

34. Krawczyk B. Learning from imbalanced data: open challenges and future directions. Prog. Artif. Intell. Springer Berlin Heidelberg; 2016; 5(4): 221–232. Available at: DOI:10.1007/s13748-016-0094-0

35. Zhou ZH. Ensemble methods: Foundations and algorithms. Ensemble Methods: Foundations and Algorithms. 2012. 1–218 p. Available at: DOI:10.1201/b12207 (Accessed: 31 January 2019)

36. Galar M., Fernandez A., Barrenechea E., Bustince H., Herrera F. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews. 2012; 42(4): 463–484. Available at: DOI:10.1109/TSMCC.2011.2161285

37. Lu W., Li Z., Chu J. Adaptive Ensemble Undersampling-Boost: A novel learning framework for imbalanced data. J. Syst. Softw. Elsevier Inc.; 2017; 132: 272–282. Available at: DOI:10.1016/j.jss.2017.07.006

38. Yuan X., Xie L., Abouelenien M. A regularized ensemble framework of deep learning for cancer detection from multi-class, imbalanced training data. Pattern Recognition. 2018; 77: 160–172. Available at: DOI:10.1016/j.patcog.2017.12.017

39. Sun J., Lang J., Fujita H., Li H. Imbalanced enterprise credit evaluation with DTE-SBD: Decision tree ensemble based on SMOTE and bagging with differentiated sampling rates. Information Sciences. 2018; 425: 76–91. Available at: DOI:10.1016/j.ins.2017.10.017

40. Feng W., Huang W., Ren J. Class imbalance ensemble learning based on the margin theory. Applied Sciences (Switzerland). 2018; 8(5). Available at: DOI:10.3390/app8050815

41. Feng W., Huang W., Ren J. Class Imbalance Ensemble Learning Based on the Margin Theory. Applied Sciences. 2018; 8(5): 815. Available at: DOI:10.3390/app8050815

42. Zhou ZH. Ensemble methods: Foundations and algorithms. Ensemble Methods Found. Algorithms. 2012. 1–218 p. Available at: DOI:10.1201/b12207

43. Liu XY., Wu J., Zhou ZH. Exploratory under-sampling for class-imbalance learning. Proc. - IEEE Int. Conf. Data Mining, ICDM. IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS; 2006. pp. 965–969. Available at: DOI:10.1109/ICDM.2006.68

44. Schapire RE. A brief introduction to boosting. IJCAI International Joint Conference on Artificial Intelligence. 1999. pp. 1401–1406. Available at: DOI:citeulike-article-id:765005 (Accessed: 27 January 2019)

45. Vluymans S., Triguero I., Cornelis C., Saeys Y. EPRENNID: An evolutionary prototype reduction based ensemble for nearest neighbor classification of imbalanced data. Neurocomputing. 2016; 216: 596–610. Available at: DOI:10.1016/j.neucom.2016.08.026

46. Le T., Vo MT., Vo B., Lee MY., Baik SW. A Hybrid Approach Using Oversampling Technique and Cost-Sensitive Learning for Bankruptcy Prediction. Complexity. 2019; 2019: 1–12. Available at: DOI:10.1155/2019/8460934

47. David Dangut M., Skaf Z., Jennions I. Rescaled-LSTM for Predicting Aircraft Component Replacement Under Imbalanced Dataset Constraint. 2020 Advances in Science and Engineering Technology International Conferences (ASET). IEEE; 2020. pp. 1–9. Available at: DOI:10.1109/ASET48392.2020.9118253

48. Lee J., Lee YC., Kim JT. Fault detection based on one-class deep learning for manufacturing applications limited to an imbalanced database. Journal of Manufacturing Systems. Elsevier Ltd; 2020; 57(November): 357–366. Available at: DOI:10.1016/j.jmsy.2020.10.013

49. Masnadi-Shirazi H., Vasconcelos N. Cost-Sensitive Boosting. {IEEE} Trans. Pattern Anal. Mach. Intell. 2016; 33(2): 294–309.

50. Hastie T., Tibsshirani R., Friedman J. The Elements os Statistical Learning. 2nd edn. New York, New York, USA: Springer US; 2009. 1–656 p. Available at: DOI:10.1007/b94608

51. Friedman J., Hastie T., Tibshirani R. Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). The Annals of Statistics. 2000; 28(2): 337–407. Available at: DOI:10.1214/aos/1016218223

52. Kull M., Silva Filho TM., Flach P. Beyond Sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. Electronic Journal of Statistics. 2017; 11(2): 5052–5080. Available at: DOI:10.1214/17-EJS1338SI

53. Zadrozny B., Elkan C. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. Icml. 2001; : 1–8. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.3039&rep=rep1&type=pdf

54. Dal Pozzolo A., Caelen O., Bontempi G., Johnson RA. Calibrating Probability with Undersampling for Imbalanced Classification Fraud detection View project Volatility forecasting View project Calibrating Probability with Undersampling for Imbalanced Classification. 2015; Available at: DOI:10.1109/SSCI.2015.33

55. Guo H. Learning from Imbalanced Data Sets with Boosting and Data Generation : The DataBoost-IM Approach. 6(1): 30–39.

56. Liu X-Y., Wu J., Zhou Z-H. Exploratory Undersampling for Class Imbalance Learning. IEEE Transactions on Systems, Man and Cybernetics. 2009; 39(2): 539–550. Available at: DOI:10.1109/TSMCB.2008.2007853

57. Masnadi-Shirazi H., Vasconcelos N. Cost-Sensitive Boosting. {IEEE} Trans. Pattern Anal. Mach. Intell. 2011; 33(2): 294–309.

58. Lee DH., Yang JK., Lee CH., Kim KJ. A data-driven approach to selection of critical process steps in the semiconductor manufacturing process considering missing and imbalanced data. Journal of Manufacturing Systems. Elsevier; 2019; 52(August 2018): 146–156. Available at: DOI:10.1016/j.jmsy.2019.07.001

5.2 Aircraft Predictive Maintenance Modeling using a Hybrid Imbalance Learning Approach

The continued development of the industrial internet of things (IIoT) has caused an increase in industrial datasets' availability. The massive availability of assets operational datasets has prompted more research interest in the area of condition-based maintenance towards the API-led integration for assets predictive maintenance modelling. The large data generated by industrial processes inherently comes along with different analytical challenges. Data imbalance is one of such problems that exist in datasets. It affects the performance of machine learning algorithms, which yields imprecise predictions. This paper proposes an advanced approach to handling imbalance classification problems in equipment heterogeneous datasets. The technique is based on a hybrid of soft mixed Gaussian processes with the EM method to improve the prediction of the minority class during learning. The algorithm is then used to develop a prognostic model for predicting aircraft component replacement. We validate the feasibility and effectiveness of our approach using real-time aircraft operation and maintenance datasets. The dataset spans over seven years. Our approach shows better performance compared to other similar methods.

5.2.1 Introduction

The recent advancement in industrial technology, known as the fourth industrial revolution, has broken the barriers between the physical and digital worlds. The technological revolution involves the integration of technologies such as the Internet of Things (IoT), the application of artificial intelligence (AI), the Application Programming Interface (API), and machine learning in the industrial process to enhance productivity. This collective force has brought an increase in the generation and availability of industrial datasets. Businesses are leveraging the large available datasets generated by the modern industrial system to make a more informed decisions.

One such application area is the vehicle’ predictive maintenance, instead of relying on average life statistics. It uses direct condition monitoring data to forecast or estimate upcoming maintenance based on historical knowledge. Predictive maintenance has a comparative advantage in almost all industries compared to other forms of maintenance strategies. Application of equipment prognostics is vital, especially in a domain where the system's criticality or component may affect health and safety, such as aircraft health monitoring, nuclear industries, and many more.

The increasing availability of datasets also comes along with more analytical challenges, which raises the necessity of applying an advanced algorithm to harness knowledge for better-informed decisions. This necessity is highlighted in equipment' predictive maintenance, where the monitoring system is expected to provide accurate prognostic alerts in advance to plan for maintenance ahead of time to avoid unexpected failure. One of the analytical challenges that inherently comes with raw asset operational datasets and affects data-driven predictive models' performance is the data imbalance problem. The data Imbalance problem is a well-known problem in machine learning and data mining communities [1]. Most real-world applications face a common problem because industrial processes are designed to function normally with few faults recorded. Data imbalance occurs in industrial datasets due to a rare event failure compared to the healthy state of the monitoring system.

Rare failure occurs due to the infrequent occurrence of some unexpected equipment break down, causing unplanned maintenance. For example, in aircraft scheduled maintenance strategy, the failure that occurs between scheduled maintenance-defined time intervals is rare. Still, their impact on business can be grave [2,3]. Therefore, rare failures are often more critical to predict because their occurrence could negatively impact society or business [4]. The majority of the data generated from the aircraft central maintenance system is highly characterised by a healthy majority and a faulty minority (represents the rare failures).

Furthermore, an extreme data imbalance problem is a scenario where a dataset contains a high representation of samples in one class than other classes present in a dataset. Learning from an extremely imbalanced dataset is quite challenging for traditional machine learning algorithms, which often leads to undesired prediction outcomes. The class Imbalance problem has been shown to degrade predictive modelling performance, causing imprecise prediction [5]. In a situation where the imbalance ratio is extreme, the learning algorithm may sometimes consider the minority class as an outlier or noise and drop them, resulting in bias learning from one class [6].

The class imbalance problem has recently drawn significant research attention. A lot of techniques and approaches for handling imbalance problems have been proposed in the literature. The majority of these techniques are based on the nature (distribution) of the dataset or its application domain. Although Imbalance learning has been extensively researched [7][1], open literature lacks a unified solution to handling the imbalanced dataset for predictive maintenance modelling, especially in the aerospace domain, particularly aircraft central maintenance system dataset. Hence, it is still an open area of research.

Therefore, this paper proposed a hybrid technique to overcome the extreme imbalance problem in heterogeneous datasets. The proposed approach comprises the integration of boosting with divide and merge strategy and Mixed Gaussian Process (MPG). The technique is designed to enhance predictive maintenance modelling for aerospace applications. The focus is on enhancing the prediction of the minority class in the process of developing an aircraft components failure prognostic model.

This paper presents the following contributions.

1. A proposed hybrid technique for improving the prediction of the minority class in the imbalanced dataset is designed and implemented.

2. A predictive model for predicting component replacement is developed to improve predictive maintenance in aerospace.

3. The model is validated on real-time aircraft operational and maintenance dataset.

5.2.2 Related Work

Many approaches and methods for handling imbalanced datasets in the process of developing data-driven predictive modelling have been proposed in the literature. A comprehensive review of the existing methods can be found in [4,8,9]. The methods can be summarised into three main categories: 1. Data level approach:- It involves resampling the dataset before presenting it as input to the learning algorithm, and this can be achieved in different ways, some of which are under-sampling (that is, randomly taking out some samples from the majority class to balance with the minority class). Over-sampling (involves adding more samples to the minority class to adjust with the majority class). A hybrid of Under-sampling and Over-sampling is possible.

The algorithm level approach involves modifying the learning algorithm to respond favourably to both classes during learning. A typical example is cost-sensitive learning. The weight of classification is defined for each class; for example, a higher weight can be set for a minority class so that during learning, the algorithm will focus more on the minority class improving its prediction.

Another approach is the ensemble and hybrid methods; this approach involves the combination of two or more approaches to improve the predictive performance of the machine learning model.

The aforementioned approaches have their pros and cons. For instance, in the data under-sampling methods, since samples are reduced from the majority class, it makes it prone to losing informative data points, which could be used in defining a decision boundary during learning. Similarly, in the oversampling approach, since artificial synthetic data points are created, this can lead to generalisation, and the original structure of the dataset is altered, which can affect the output of the model. Likewise, In the algorithm level, cost-sensitive methods, defining the cost of misclassification for each class is quite challenging. Therefore, because of our dataset's peculiarity and the application domain, none of the out-of-box existing solutions was suitable.

5.2.2.1 Machine Learning

Machine learning is grouped into different types: supervised learning, unsupervised learning, semi-supervised learning, active learning, and reinforcement learning. The use of more than one type of learning is referred to as hybrid learning. In this study, we use classification, supervised learning, and clustering, which is unsupervised learning, hence the hybrid. In supervised learning, the algorithm builds a mathematical model from a dataset that contains input and known output (labels). The conventional approaches are classification and regression. In the case of unsupervised learning, the algorithm builds a mathematical model from the dataset that contains only input variables without labels. Unsupervised learning is mostly used to find structures in the dataset, such as grouping or clustering[10].

Many machine learning algorithms exist; their application depends on the nature and type of dataset and the problem at hand. For example, Support Vector Machine (SVM) and Decision Tree (DT) algorithms can be used for classification in supervised learning. Likewise, K-means, Gaussian process algorithm can be used for clustering in unsupervised learning. Combining more than one weak classifier to form a robust classifier to achieve a better result is known as ensemble learning [11][12]. Many recent machine learning approaches have been designed based on ensemble learning to deal with various categories and dimensions of data imbalance challenges. Also, in many application domains [13], the most common ensemble learning techniques are bagging and boosting [14].

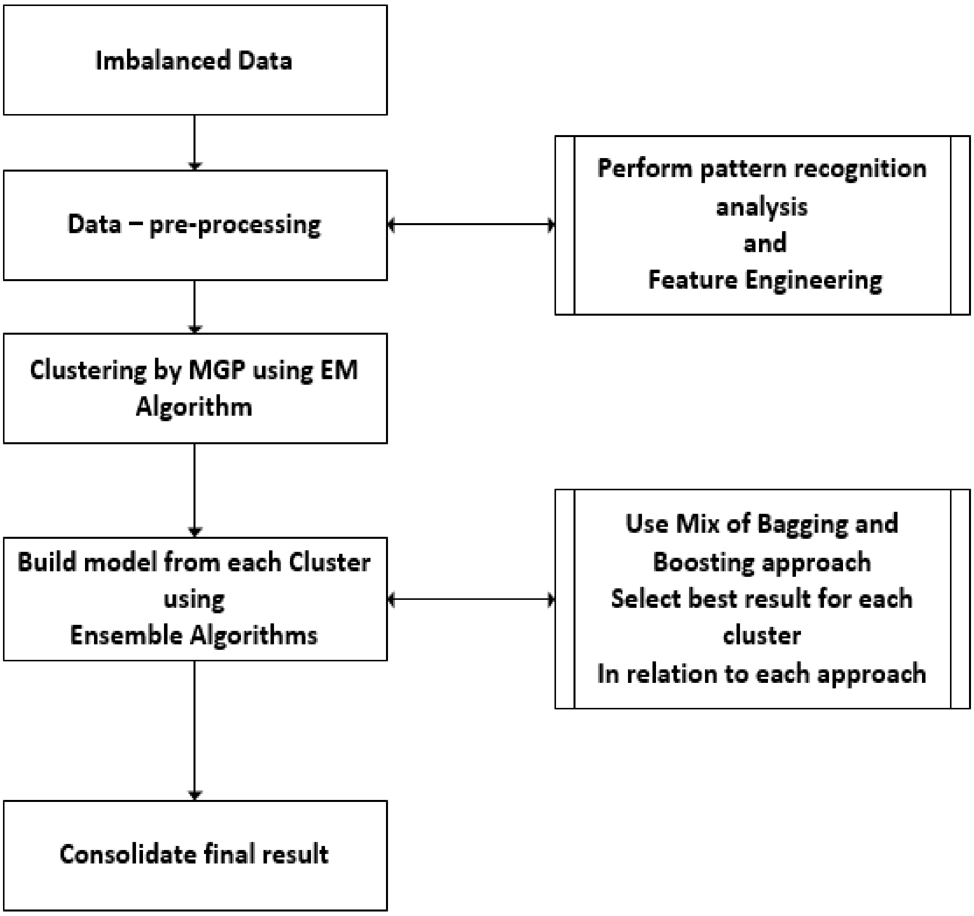

Imbalanced Data
Data – pre-processing
Perform pattern recognition analysis and Feature Engineering
Clustering by MGP using EM Algorithm
Build model from each Cluster using Ensemble Algorithms
Use Mix of Bagging and Boosting approach Select best result for each cluster In relation to each approach
Consolidate final result

Figure 5- 9 Machine Learning Hybrid Framework for enhancing class prediction

Figure 5-8 shows the Machine Learning Hybrid Framework for enhancing class prediction. The framework is based on the hybrid approach. It combines supervised and unsupervised machine learning methods to improve the prediction of the minority class.

5.2.2.2. Mixed Gaussian Process Methods

A study presented by Vandaplas et al. [15] and Fong et al. [16] shows that the clustering method, which is based on learning a mixture of Gaussians, involves collecting a mix of k component distribution to form a mixture distribution function. $f\ (x) = \sum_{k=1}^{k} \alpha_k f_{k\ (x)}$ (5- 7)

α_k is the mixing weight for the Ith component in the construction of Gaussians distribution $f(x)$. K is the number of component distribution

The dataset used in this study is multi-variant, and some variables' distribution is unknown. We use the Expected Maximisation algorithm (EM) to minimise a likelihood function by iterating and guessing the distribution until convergence. K-means algorithm groups data using a hard clustering approach with no overlapping of clusters. (Point belongs to a cluster, or it does not belong to) While the EM algorithm computes the probability that it belongs to a cluster, which is referred to as soft clustering [17][18]. Figure 5-9 shows the clustering method based on learning a mixed of Gaussians.

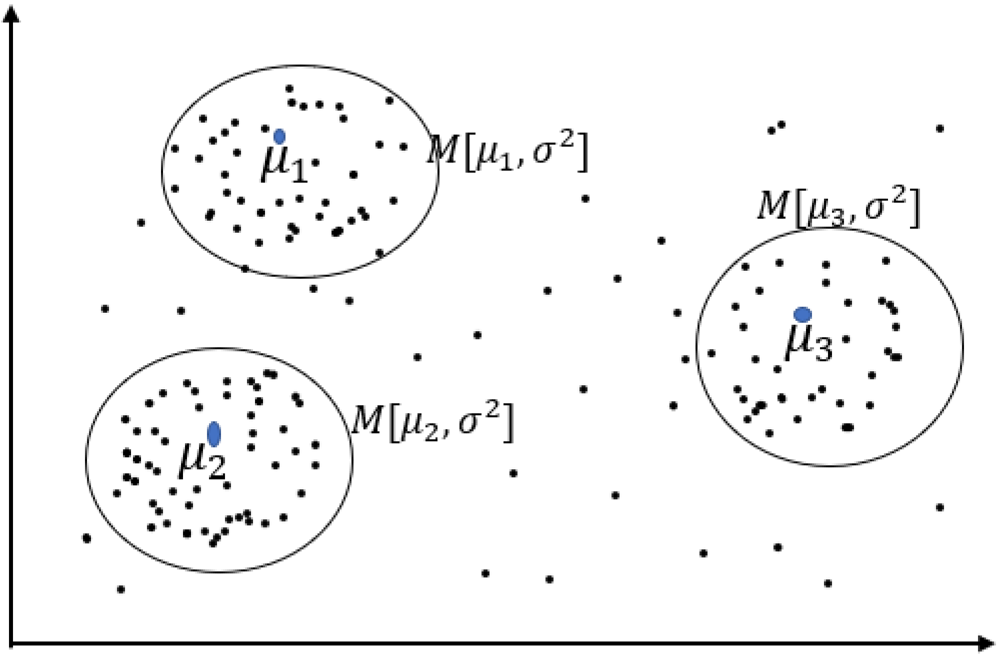


Figure 5- 10 Clustering method based on learning a mixed of Gaussians [16]

μ_i: is the mean; that is centre of the mass

σ^2: is the variance; that is spread of the mass

```
Given an unknown observation of $\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \dots \mathbf{x_n}$
1. Start with two randomly placed Gaussians ( $\boldsymbol{\mu_1}, \boldsymbol{\delta_1^2}$ ),
   ( $\boldsymbol{\mu_2}, \boldsymbol{\delta_2^2}$ ) in the space
2.  E- Step:
        For each point: $\mathbf{P(1|x_i)}$ = does it look like it came
           from 1?
3. M-step:
        Adjust ( $\boldsymbol{\mu_1}, \boldsymbol{\delta_1^2}$ ) and ( $\boldsymbol{\mu_2}, \boldsymbol{\delta_2^2}$ ) to fit points, assign
           to them
4. Loop until convergence
```

Figure 5- 11 EM algorithm 1

5.2.3 Methodology

Our proposed approach is similar to the hybrid method algorithm proposed by VanderPlas et al. [15]. However, our approach differs in the base learning algorithm. Instead of using hard K-means for clustering, we use a soft Mixed Gaussian Process with EM (MGP-EM).

The MGP-EM approach helps in computing the probability of points belonging to the cluster, which deals with an in-between point to avoid ambiguity problems in clustering. The proposed method is designed to overcome the problem of class-overlapping or small-size samples, which is difficult for the classifier to learn, hence improving the prediction of a minority class. It is also to handle the problem of over-sampling using K-means clustering, which is sensitive to outliers and noise and unable to handle more massive datasets. Putting the data into lagging windows and bootstrapping helps in the learning phase by keeping the statistics, which avoids processing the whole dataset; instead, it keeps only the statistics of each window's outcome.

Bagging-based (i.e., divide and merge) improves model performance, increases detection rate (True Positive), and reduces the false positive. The mixed Gaussian process is used as a based learner in the Boosting step.

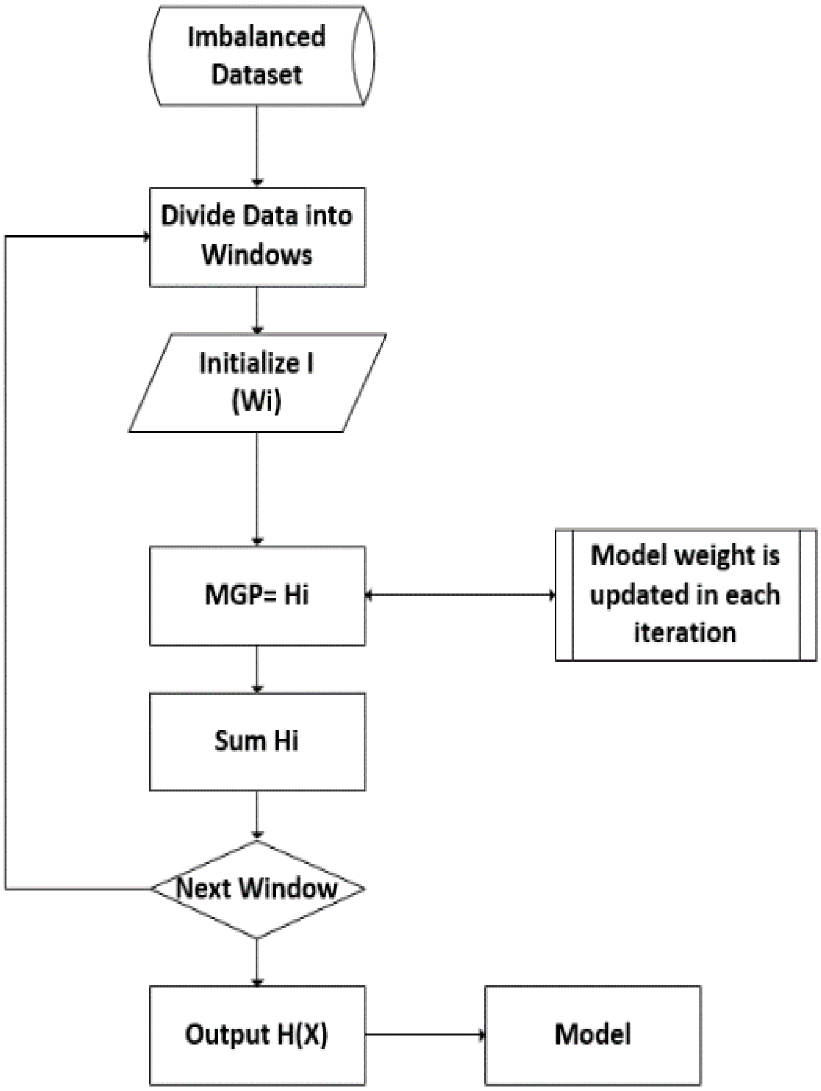


Figure 5- 12 The Architecture of the proposed approach

We performed cross-validation during the training phase to avoid model over-fitting problems. We classify the model using the proposed hybrid method, using a cluster-based –Mixed Gaussian Process, as weak learners. The result of MGP-EM is then combined with the Ensemble bagging method using the random forest as a based learner.

```
STEP 1: Input the Imbalanced Dataset D = $x\epsilon \{x_{1,} x_{2,} \ldots x_n \}$
STEP 2: Divide the Data in Windows $W1, W2$ ... $Wn$
STEP 3: Initialisation $x_1 = 1/n$
STEP 4: Then Mixed Gaussian Process (EM) is used as a base learner in the boosting = $\alpha_k f_k (x)$ and adjust weight
STEP 5: Calculate the True Positive and False Positive Rate
STEP 6: Iterate Until the end of windows
STEP 7: return final hypothesis $H(x) = \sum_{k=1}^{k} \alpha_k f_k (x)$
STEP 8: END
```

Figure 5- 13 Hybrid Algorithm 2

5.2.4. Experimental Setup

To validate the effectiveness of the proposed approach. The experiment uses a dataset obtained from a fleet of commercial aircraft, which has been recorded for over seven years. The data is a recorded component failure recoded as a log by aircraft cental maintenance computers. The data is heterogeneous, meaning it comes from different aircraft sub-systems, and it contains numerical, textual, and symbolic.

As a first step, the data is preprocessed and transformed for machine learning because to make use of the log-based dataset for developing a predictive model, and the log needs to be filtered and interpreted and predictive feature extracted.

The data is then divided into two using the event date. Data from 2011 to 2015 is used for training the model and from 2016 to 2018 for testing the model.

In the experiment, we investigate the performance of the proposed method against existing ensemble learning methods.

We measured the performance of the model using precision, recall, F1-score.

We presented the experimental results in Table 5-3. In the experiment, we select out of many the aircraft components identified by Functional Item Number (FIN) that are replaced due to an unplanned breakdown. We focused on the aircraft component with the highest number of replacements in the dataset. The components considered are 4000KS - Electronic Control Unit, 4000HA - High-Pressure Bleed Valve, 4001HA – Pressure Regulating Valve, 5RV1 – Satellite Data unit.

5.2.5. Result and Discussion

As seen in Table 5-3, The proposed method's result is compared against the baseline algorithm, which is the Random Forest algorithm (RF). The result shows that our approach outperformed the baseline method both in precisions and recall. Similarly, the F1-score indicates that the proposed approach is able to detect both classes with less bias. The high recall score shows that the proposed approach is able to detect the minority class better, which is our class of interest (the rare faults). However, the

Table 5- 3 The result showing the performance of the proposed Framework

	Ensemble method -Random Forest (Baseline)				**Proposed Hybrid Approach**			
Components	4000KS	4000HA	4001HA	5RV1	**4000KS**	**4000HA**	**4001HA**	**5RV1**
Precision	0.77	0.70	0.71	0.79	**0.94**	**0.90**	**0.92**	**0.96**
Recall	0.60	0.59	0.60	0.63	**0.85**	**0.80**	**0.82**	**0.89**
F1-Score	0.67	0.64	0.65	0.70	**0.89**	**0.85**	**0.87**	**0.93**
AUC	0.60	0.65	0.66	0.72	**0.90**	**0.86**	**0.88**	**0.95**
IR	0.0031	0.0024	0.0028	0.0039	**0.0031**	**0.0024**	**0.0028**	**0.0039**

model includes some points of the majority (false negatives). This can be considered acceptable in this context, as we are more interested in reducing the false-positive rate than a false negative. It can also be observed that the imbalance ratio has an effect on the result. In the cases with higher IR, the model is able to learn better, while in the cases with lower performance, the performance is dropped. Despite the extreme imbalance ratio in all the cases considered, the proposed method was able to predict more than 80% of the rare equipment failure. The result also shows the effectiveness of the model in handling extreme class imbalance problems in big data.

5.2.6 Conclusion

This paper proposes a hybrid framework for data-driven predictive maintenance. We focus on enhancing the prediction of the minority class in the data Imbalance classification problem. The data imbalance problem is a data analytics challenge that degrades the performance of data-driven predictive models. Our approach is based on a hybrid ensemble method, which improves the prediction of the minority class during learning. The proposed MGP-EM approach helps in computing the probability of points belonging to the cluster, which deals with an in-between point to avoid ambiguity problems in clustering. The proposed method overcomes the problem of class-overlapping or small-size samples, which is difficult for the classifier to learn, hence improving the prediction of a minority class. It also overcomes the problem of over-sampling using K-means clustering, which is sensitive to outliers and noise and unable to handle more massive datasets. In the feature, we will try to improve the aircraft's predictive model performance by including other aircraft-related datasets such as environmental and weather data. We will also work on improving the detection of extreme minorities in a multi-class context by applying the deep-learning approach.

5.2.7 References

1. He H., Garcia EA. Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering. 2009; 21(9): 1263–1284. Available at: DOI:10.1109/TKDE.2008.239

2. Dangut M david., Skaf Z., Jennions IK. An integrated machine learning model for aircraft components rare failure prognostics with log-based dataset. ISA Transactions. Elsevier Ltd; 2020; (xxxx). Available at: DOI:10.1016/j.isatra.2020.05.001

3. Wang Y. Strategies for Aircraft Using Model-Based Prognostics. 2018;

4. Shang J., Mingyun G., Yijing L., Bing G., Yuanyue H., Haixiang G. Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications. Elsevier Ltd; 2016; 73: 220–239. Available at: DOI:10.1016/j.eswa.2016.12.035

5. Dangut MD., Skaf Z., Jennions IK. An integrated machine learning model for aircraft components rare failure prognostics with log-based dataset. ISA Transactions. Elsevier Ltd; 2020; (xxxx). Available at: DOI:10.1016/j.isatra.2020.05.001

6. David Dangut M., Skaf Z., Jennions I. Rescaled-LSTM for Predicting Aircraft Component Replacement Under Imbalanced Dataset Constraint. 2020 Advances in Science and Engineering Technology International Conferences (ASET). IEEE; 2020. pp. 1–9. Available at: DOI:10.1109/ASET48392.2020.9118253

7. Branco P., Torgo L., Ribeiro RP. A Survey of Predictive Modelling under Imbalanced Distributions Adapting Resampling Strategies for Dependency-Oriented Data in Imbalanced Domains View project International Workshop on Cost-Sensitive Learning View project A Survey of Predictive Modelling . 2015. Available at: https://www.researchgate.net/publication/275968092

8. He H. Imbalanced Learning. Self-Adaptive Systems for Machine Intelligence. New Jersey: John Wiley & Sons, Inc.,Hoboken, New Jersey.; 2011. 44–107 p. Available at: DOI:10.1002/9781118025604.ch3

9. Rout N., Mishra D., Mallick MK. Handling imbalanced data: A survey. Advances in Intelligent Systems and Computing. 2018. 431–443 p. Available at: DOI:10.1007/978-981-10-5272-9_39

10. Abraham A., Pedregosa F., Eickenberg M., Gervais P., Muller A., Kossaifi J., et al. Hands-On Machine Learning with Scikit-Learn and TensorFlow.pdf. O'Reilly Media; 2014. 568 p. Available at: DOI:10.3389/fninf.2014.00014

11. Camacho-Navarro J., Ruiz M., Villamizar R., Mujica L., Moreno-Beltrán G. Ensemble learning as approach for pipeline condition assessment. Journal of Physics: Conference Series. 2017. Available at: DOI:10.1088/1742-6596/842/1/012019

12. Zhang D., Jiao L., Bai X., Wang S., Hou B. A robust semi-supervised SVM via ensemble learning. Applied Soft Computing Journal. 2018; 65: 632–643. Available at: DOI:10.1016/j.asoc.2018.01.038

13. Polikar R. Ensemble based systems in decision making. IEEE Circuits and Systems Magazine. 2006; 6(3): 21–44. Available at: DOI:10.1109/MCAS.2006.1688199

14. Zhou ZH. Ensemble methods: Foundations and algorithms. Ensemble Methods: Foundations and Algorithms. 2012. 1–218 p. Available at: DOI:10.1201/b12207 (Accessed: 31 January 2019)

15. VanderPlas J. Python Data Science Handbook. O'Reilly. 2016. p. 541. Available at: http://shop.oreilly.com/product/0636920034919.do%0Ahttps://jakevdp.github.io/PythonDataScienceHandbook/05.01-what-is-machine-learning.html

16. Fong Chun Chan. Using Mixture Models for Clustering. Available at: http://tinyheero.github.io/2015/10/13/mixture-model.html (Accessed: 26 May 2019)

17. Batista GEAPA., Prati RC., Monard MC. Balancing Strategies and Class Overlapping. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2010. 24–35 p. Available at: DOI:10.1007/11552253_3

18. Visa S., Ralescu A. Learning imbalanced and overlapping classes using fuzzy sets. Workshop on Learning from Imbalanced Datasets II (ICML ’03). 2003; (0): 91–104. Available at: https://link.springer.com/chapter/10.1007/978-3-540-24694-7_32

CHAPTER 6: Deep Learning Approach for Rare Failure Prediction

This chapter presents a new deep-learning technique for handling extreme rare failure predictions. Two new proposed methods are implemented: A novel approach based on combining two deep learning techniques, auto-encoder (AE) and Bidirectional Gated Recurrent Unit (BGRU) networks. A loss function is derived for deep neural networks, enabling the deep learning algorithms to respond favourably to minority and majority groups. The derived loss function is implemented using a rescaled-LSTM. The chapter is organised as follows:

6.1 A Rare Failure Detection Model for Aircraft Predictive Maintenance Using a Deep Hybrid Learning Approach

The Aircraft Central Maintenance System (ACMS) log records are a potential data source for developing data-driven predictive models, which can unlock several benefits for the aircraft health monitoring system and condition-based maintenance. However, developing data-driven models using ACMS data faces many challenges, such as the low representation of failure behaviour related to target component failure, creating a skewed distribution in the datasets. The rare representation of failure results in what is known as an imbalanced classification problem in machine learning. Training any traditional machine learning algorithm with an extremely imbalanced dataset can cause biases in a model. Thus, this study presents novel deep learning techniques based on the auto-encoder (AE) and Bidirectional Gated Recurrent Unit (BGRU) networks to handle extremely rare failure prediction in aircraft' predictive maintenance modelling. AE is modified and trained to detect rare failures, and the result from AE. is fed into the convolutional neural network and bidirectional gated recurrent unit CNN-BGRU network to predict the next occurrence of failure. This hybrid approach helps in addressing the imbalance problem during model training. The effectiveness of the proposed method is evaluated using real-world test cases of log-based warning and failure messages obtained from the ACMS fleet database and maintenance history records. The AE-CNN-BGRU model is compared with other similar deep learning methods. The results show improved performance with 25% better precision, 14% recall, and 3% G-mean. It also indicates robustness in predicting rare failure within a defined, useful period.

6.1.1 Introduction

It is important to note that this study is an extension of work presented in the 4th IFAC Workshop on Advanced Maintenance Engineering, Services and Technologies (AMEST 2020) [1].

Unscheduled aircraft maintenance can cause flight cancellation or delay due to the unavailability of spares at the failure location. It can result in unwanted downtime, which increases the airlines' operational costs. Reducing the number of unscheduled maintenance activities through predictive modelling is an excellent initiative for airlines; it reduces maintenance costs and increases fleet availability. According to Airbus [2], by 2025, unscheduled aircraft grounding for fault repairs could cease due to data analytics and operational experience. Aircraft health monitoring and Predictive maintenance could enhance the elimination of unscheduled groundings of aircraft by systematically scheduling maintenance intervals more regularly to avoid Aircraft on Ground (AOGs) and the associated operational interruptions[3] [2]. A good predictive model could tell which aircraft parts need schedule checks and those that don't need it but achieving such maintenance accuracy necessities experience and the right technology [2].

The recent advancement in artificial intelligence (AI) and other related technologies such as the Internet of Things (IoT), machine learning, and symbolic reasoning is causing a paradigm shift in every aspect of human life, including manufacturing, transportation, energy, advertisement. Aerospace is one of the industries that has mostly been transformed through the application of AI technologies. Mainly aircraft maintenance is rapidly leveraging AI to develop predictive maintenance towards "aircraft smart-maintenance". The advancement towards smart maintenance solutions is a situation where the machine learning algorithms are trained to predict failure and provide possible actions based on the predicted failure. The conditioned-based predictive maintenance provides cost-saving over time-based preventive maintenance Burijs et al.[4] as maintenance is done based on the condition of the component, not time-based as in preventive maintenance. The large amount of data generated from IoT devices installed in aircraft to monitor various components' health conditions combined with data analytics through machine learning can significantly improve aircraft maintenance activities.

Applying correct data analytics and training machine learning algorithms with a vast amount of data can reveal underlying patterns and trends that are not visible to humans. The information discovered can support proactive decision-making, such as recommending the best maintenance actions. Therefore, well-develop machine algorithms are needed to harness relevant information from big data. As Artificial Intelligence (AI) and related technologies continue to advance, data becomes more available with a less challenging acquisition, storage, and processing methods. However, newer analytical challenges are emerging. One unique challenge is the extremely rare event prediction, which is when events are infrequent, causing the generated data to be imbalanced, meaning there are significantly fewer data in one class compared to other classes. Training a traditional machine learning algorithm with a skewed dataset has been shown to degrade the resulting model's performance [4]. Therefore, to develop a robust machine learning model for predictive maintenance, it is vital to address imbalanced data before training (data level approach) or to train the model (algorithm level approach).

The challenge traditional machine learning algorithms face with the extremely imbalanced dataset is that they are built on the assumption that the data distribution is always balanced, and the cost of misclassification is the same for all classes [5]. In reality, that assumption is untrue because there are some domains where the data is highly imbalanced, and the cost of misclassification is high. An example of such a domain is log data generated by the aircraft central maintenance system known as ACMS data. ACMS data is imbalanced because aircraft component failure rarely occurs during regular flight operations due to robust safety measures. The generated ACMS data will exhibit skewness, where the majority of data represents the healthy state, and the minority represents failure. Apart from the extremely imbalanced problem, ACMS data poses several analytical issues, such as irregular patterns and trends. In this case, the standard machine learning algorithm and feature selection or extraction methods become less effective for highly imbalanced data [6]. Training machine learning algorithms with imbalanced data has been shown to degrade data-driven models' performance, causing unreliable prognostics [7], [8].

There have been recent improvements in predictive modelling research from both academic and industrial perspectives. [9]. The predictive maintenance modelling approach can be summarised into physics-based, knowledge-based, data-driven-based, and hybrid-based approaches. The physics-based approach focuses on the equipment degradation process, and it requires an understanding of the components' underlying physical failure mechanisms [10]. The physics-based modelling approach's application can be seen in [11], [12], where a digital model of equipment is created to enable the Digital Twin (D.T.) concept in predictive maintenance applications. DT is the concept where multi-physics modelling, together with data-driven analytics. GE has developed an intelligent IoT-based monitoring and diagnostics platform based on DT to predict physical asset future [13]. The advantage of this approach is it is applicable even if the dataset is scarce.

Another approach to predictive maintenance modelling is knowledge-based or expert system modelling. This approach involves a combination of domain expert knowledge and computational intelligence techniques. It stores information from domain experts, and rulesets are defined based on the knowledge base for interpretation [14]. The knowledge-based approach has been applied for predictive aircraft maintenance [15], [16]. The authors develop a framework and design methodology for the development of knowledge-based condition monitoring systems. In practice, knowledge-based approaches are more useful for a small and simple system. Its application in a large and complex system is quite challenging and, in some cases, impractical because domain experts need to continually update the rules in the event of any upgrades or changes, which is cumbersome.

The data-driven approach involves learning systems behaviour directly from already collected historical operational data to predict the future of a system's state or identify and match similar patterns in the dataset to infer Remaining Useful Life (RUL) or other insights. The data-driven modelling methods can be grouped into Artificial Intelligence (AI) based, statistical modelling methods, and sequential pattern mining modelling methods [17]. AI methods include machine learning, Bayesian methods, and deep learning methods. AI-based methods have been widely used for developing predictive maintenance models in different industries. Çinar et al. [19] provided a detailed survey on recent applications of AI in Predictive Maintenance. The hybrid approach includes a combination of two or more techniques for estimation to improve accuracy. Improving accuracy in rare failure prediction requires a robust hybrid approach. In recent times, deep learning (DL) models have been shown to produce state-of-art performance when trained with large datasets [18], [19] because of their capability of combining feature extraction with learning. The advances in machine learning research, especially using deep neural networks to learn more complex temporal features, make DL suitable for a large log-based dataset. Other work has shown the effectiveness of DL models in handling extremely imbalanced datasets, especially using log-based ACMS datasets to develop aircraft predictive maintenance models [9].

In this study, a data-driven model is proposed for rare failure prediction. The model consists of deep neural networks, the auto-encoder to detect failures and bidirectional gated recurrent unit (BGRU) networks combined with Convolutional Neural Networks (CNN) to learn the co-relationships between variables, enhancing the prediction of rare failure. The model's effectiveness is evaluated using real-world log-based ACMS time series data. The proposed model will help mitigate the effects of unscheduled aircraft maintenance, producing systematic conditioned-based predictive maintenance, a step towards a smart-aircraft maintenance system.

The remainder of this paper is structured as follows. Section 6.1.2 discusses the related work. Section 6.1.3 provides a methodology that shows a detailed architecture of the auto-encoder, CNN, and BGRU. Section 6.1.4 presents the experimental setup and case study. The experimental result is presented and discussed in section 6.1.5. Finally, section 6.1.6 presents the conclusion and further work.

6.1.2 Related Work

Deep learning is a branch of machine learning consisting of numerous processing layers that learn data representations at multiple levels of abstraction using artificial neural networks (ANN). Deep learning models have vastly enhanced the state-of-the-art performance of models in a variety of disciplines, including large-scale data processing and image identification, and many more [7]. The success has been attributed to an increase in the availability of data, hardware, and software improvements, many breakthroughs in algorithm development that speed up training and other data generalisations [20]. Despite the advances, little work has been done to investigate the effect of extremely imbalanced, class overlapping, and small class disjunct on the networks architectures. Many researchers have agreed that the subject of imbalanced data with deep learning is understudied [21]–[24]. In deep learning, the ANNs are trained to find complex structures in a dataset by using a back-propagation algorithm. The algorithm calculates errors made by the model during training, and the models' weights are updated in proportion to the error. The drawback of this learning method is that examples from both classes are treated the same. In that situation where the data is imbalanced, the model will be adapted more to the majority class than the minority class, which can affect the performance of the models [20].

The majority of the deep learning methods for imbalanced classification have depended on integrating either resampling or cost-sensitive into the deep learning process. For instance, Hensman et al. [25] use random oversampling techniques to balance the data then train the balanced data using CNN. Similarly, Lee et al. [26] use random undersampling to balance the dataset for the purpose of pretraining CNN. The use of dynamic sampling to adjust the sampling rate according to the class size for training CNN was proposed by Pouyanfar et al. [27]. Buda et al. [24] investigate the effect of random oversampling, random undersampling and two-face learning across many imbalanced datasets on deep neural networks. The literature review [20], [28] reveals that most of the proposed deep learning resampling approaches for imbalanced problems use image datasets and CNN architecture. The need to Investigate the effect of imbalanced on other deep learning architectures and to use time-series is still lacking.

On the other hand, several studies have focused on applying cost-sensitive strategies to solve the problem of imbalanced classification, which entails changing the deep learning process to favour both classes during model training. For example, Khan SH et al. [29] proposed a cost-sensitive deep neural network that can automatically learn robust feature representations for both the majority and minority classes. Also, Zhang et al. [30] propose cost-sensitive deep belief networks, and Wang H et al. [31] propose a cost-sensitive deep learning approach to predict hospital readmission. Also, the use of loss function to control biases has been shown in Wang S et al. [6]. The authors proposed a novel loss function called mean false error and its improved version mean squared false error for learning from an imbalanced dataset. Similarly, a new loss function called Focal loss was proposed by Lin et al. [32] for dense object detection in image classification. The focal loss was proposed to specifically handle the challenge of extreme data imbalances commonly faced in object detection problems, where the foreground samples usually outnumber the background samples. Normally, this type of problem is mostly solved using the one-stage detection approach or two-stage detection. The two-stage detection usually performs at the cost of computation time compared to one-stage. Lin et al. [32] study focused on determining how the one-stage approach with fast computation time can achieve a state-of-the-art performance compared to the two-stage. Their study discovered that the main cause of performance degradation in one-stage detection is the imbalanced data problem.

The overwhelming background samples create imbalance, causing the majority class to account for most of the overall loss. To address that challenge, Lin et al. [8]. Proposed a loss function known as the focal loss (FL) which was derived from a normal binary cross-entropy loss. The FL is expressed as follows;

$$\text{Focal Loss FL}(\boldsymbol{p}_{,t}) = -\,(1 - (\boldsymbol{p}_t))^{\gamma} \boldsymbol{log}_{\mathbf{10}}\,(\boldsymbol{p}_t) \quad (6\text{- }1)$$

The new FL tries to reduce the impact that the majority of samples have on the loss by multiplying the cross-entropy loss with a modulating factor $-\,(1 - (p_t))^{\gamma}$ Where the hyperparameter γ ≥ 0 adjusts the learning rate, the negative samples are downweighed. Their implementation shows that using one-stage detection with focal loss by selecting the right learning rate outperformed the two-stage approach. The implantation method was only compared with cross-entropy and tested for imbalance problems in objection detection. The focal loss was later tested in image classification by K Nemoto et al. [33]. The authors use CNN architecture then compare the performance of focal loss and cross-entropy loss for image classification. The open literature lacks a study investigating the focal loss's effectiveness on time-series systems log-based datasets, particularly the ACMS dataset.

The identification and prediction of rare failures is an active research subject that has sparked the creation of a variety of methodologies [34]. Asset rare failure prediction is a critical issue that has been approached within various contexts, such as machine learning and statistics [17]. System log data has widely been used to develop rare failure predictive models in different domains. For example, deep learning has been used to predict rare IT software failures using a log-based dataset [35]. Panagiotis et al. [36] developed a failure event model using post-flight records. The authors formulated the model as a regression problem to approximate the risk of a target event's occurrence, using multiple instance learning schemes. Sipos et al. [37] developed a data-driven approach based on multiple-instance learning for predicting equipment failures. Evgeny [10] developed a data-driven rare failure prediction model using event matching for aerospace applications. As seen in the previous study by Maren et al. [1], one of the approaches to identifying and predicting rare failure is using an anomaly detection approach, which is framed in the form of unsupervised machine learning, where the data is divided and labelled as negative and positive samples. In the case of using an autoencoder, each class is treated separately, the negatively labelled sample's low dimensional features are extracted from higher dimension data using any feature extraction processes. Then rare failures are detected and predicted based on the reconstruction error. Most of the well-known traditional or typical data reduction and fault detection methods are the Principal Component Analysis (PCA), Partial Least Square (PLS), and Independent Component Analysis (ICA). These methods use different ways to reduce data dimensionality, and they have achieved a varying degree of success on different data distributions [38]. However, they have fundamental limitations to the non-linear features since they rely on linear techniques. Kernel tricks have been developed to convert the non-linear raw data into linear data, and examples are the KPCA [38] and KICA [39]. However, they require high computational power due to kernel function, especially if the data is large.

Deep learning (DL) has recently proven superior performance in many areas, such as image classification. Also, it has widely been used in the finance sector for the analysis of time-series data[9]. DL can also be utilised for predictive maintenance. The system installed to monitor an asset's state generates an extensive amount of time-series data. Therefore, deep learning algorithms are trained using time-series data to find patterns to predict failures. Recent developments in deep learning have made it easy for deep, complex artificial neural networks to automatically extract features from the original dataset (dimension reduction) during training [40][41]. The Auto-encoder (AE) [42] is an example of a deep neural network algorithm that has been successfully implemented for fault detection and prediction. However, it needs larger data samples and a longer processing time to achieve higher performance [43]. Advances have been made to tackle slightly rare event predictions, especially in the aerospace domain, using machine learning approaches [44][45]. Deep learning models have also been developed for rare event predictions. For example, Wu et al.[18] developed a weighted deep representation learning model for imbalanced fault diagnosis in cyber-physical systems. Their model is composed of Long Recurrent Convolutional LSTM model with a sampling policy. Also, Khanh et al.[19] developed a dynamic predictive maintenance framework based on sensor measurements. Changchang et al. [46] combine multiple DL algorithms for prognostic and health management of aircraft. In fact, Burnaev et al. [47] pointed out that many aircraft predictive maintenance solutions are built on basic threshold settings that detect trivial errors on specific components. On the other hand, the threshold-setting strategy is prone to producing high false-positive rates, which lowers model confidence.

Although the approaches mentioned above have successfully handled normal fault detection and prediction, there was a limited study about the application of deep learning models for extremely rare failure prediction, especially for predictive aircraft maintenance using the ACMS dataset. Also, developing a robust predictive model for costly rare aircraft component failure using a large log-based dataset is quite challenging because many components work together and influence each other's lifetime. Another challenge is the heterogeneous nature of the ACMS log data, including symbolic sequence, numeric time series, categorical variables, and unstructured text.

Therefore, our approach focuses on extremely rare failure prediction using log-based aircraft central maintenance system (ACMS) data. Secondly, the work also concentrates on applying a hybrid of deep learning techniques for performance optimisation. The proposed model integrates AE with BGRU and CNN to detect and predict extreme aircraft component replacement. The hybrid method is designed to address the challenge of irregular patterns and trends caused by skewed data distributaries, hence enhancing the prediction of rare failures.

6.1.3. Methodology

6.1.3.1 Autoencoder and Bidirectional Gated Recurrent Unit Network Architecture

This section presents auto-encoder, bidirectional gated recurrent unit network architecture and how these architectures are integrated to achieve better performance on large log-based, multivariate, non-linear, and time-series datasets.

The Autoencoder (AE) [48], [49] is a specific type of multi-layer feedforward neural network where the input is the same as the output neurons. AE aims to learn the original data's internal representation by compressing the input into a lower-dimensional space called latent-space representation (see Figure 6-1). It then uses the compressed representation to reconstruct the output while minimising the error for the input data. Training is done using a back-propagation algorithm with respect to the loss function. AE comprises three components: Encoder X, latent-space P, and Decoder Y. The encoder compresses the input and produces the latent representation. The decoder then reconstructs the input only using this latent representation. An encoder with more than one hidden layer is called a deep auto-encoder.

The encoding and decoding process can be represented using the equation as follows:

$$\boldsymbol{p_i = f(w_p .\, x_i + \;\; b_t)} \qquad (6\text{- }2)$$

$$\boldsymbol{y_i = g(w_y .\, p_i + \;\; b_t)} \qquad (6\text{- }3)$$

Where f(.) and g(.) are the sigmoid functions, w_i represents the weights and b_i represents biases. The following minimised loss function is used to train the model:

$$\boldsymbol{L(X, Y) = \frac{1}{2n} \sum_i^n \| \, x_i - \; y_i \|^2} \qquad (6\text{- }4)$$

Where x_i represent the observed value, y_i represent predicted values, and n represent the total number of predicted values.

Equation (6-3) helps in checking the validity of the resulting underlying feature P.

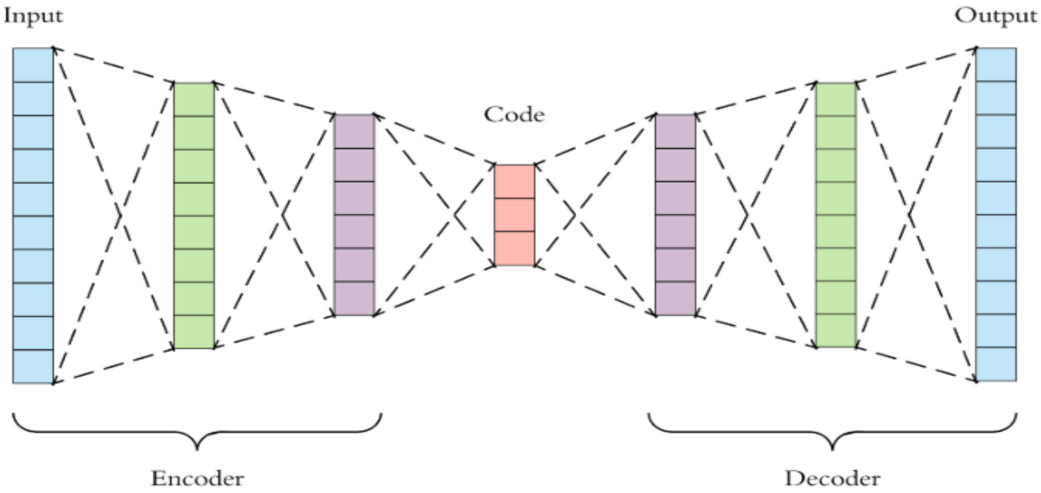


Figure 6- 1 Auto-Encoder Architecture [49]

Figure 6-1 shows a more detailed visualisation of an auto-encoder architecture. First, the input data passes through the encoder, a fully connected Artificial Neural Network (ANN), to produce the middle code layer. The decoder, which has a mirrored ANN structure, will produce the output using the middle coded layer. The goal is to get an output identical to the input. Creating many encoder layers and decoder layers will enable the AE to represent a more complex input data distribution.

6.1.3.2 The Bidirectional Gated Recurrent Unit

A Bidirectional Gated Recurrent Unit (BGRU) is a recurrent neural network that has successfully been used to solve time series sequential data problems because of its bidirectional learning approach, which enhance learning of temporal patterns in the time series data [50]. Each BGRU block contains a cell that stores information. Each block is made up of a reset and update gate, and the cells help tackle the vanishing gradient problem Janusz et al. [51]. The reset gate determines how to combine new input with previous memory, while the update gate defines how much of the previous memory to retain, BGRU comprises two GRU blocks. The input data is fed into the two networks, the feedforward and feedback with respect to time, and both of them are connected to one output layer [52]. The gates in bidirectional GRU are designed to store information longer in both forward and backward directions, providing better performance than feedforward networks. The bidirectional approach provides the capability of using both the past context and future context in a sequence. BGRU can be expressed as:

$$\boldsymbol{h_t} = [\overrightarrow{\boldsymbol{h_t},} \overleftarrow{\boldsymbol{h_t}}] \tag{6- 5}$$

$$where\ \overrightarrow{h_t,}\ is\ the\ feedfoward\ \ and\ \overleftarrow{h_t}\ the\ backward\ block$$

The final output layer at time t is:

$$\boldsymbol{y_t} = \boldsymbol{\sigma}(\boldsymbol{W_y h_t} + \boldsymbol{b_y}) \tag{6- 6}$$

Where σ is the activation function W_y is the weight, and b_y is the bias vector.

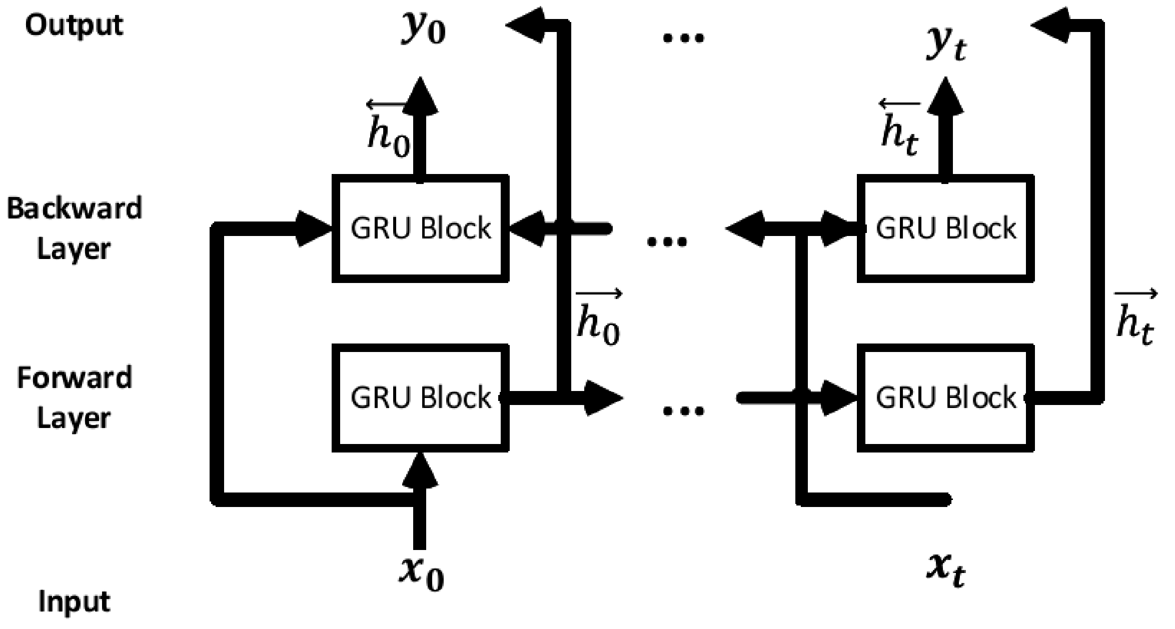


Figure 6- 2 BGRU architecture with forwarding and backward GRU layers

As seen in Figures 6-2 and 6-3, each of the GRU blocks is made up of four components. Input vector x_I with corresponding weights and bias, reset gate r_I with corresponding weight and bias $W_{r,}U_r, b_r$, update gate z_I with corresponding weight and bias $W_{z,}U_z, b_z$, and out vector h_t with its weight and bias $W_{h,}U_h, b_h$. Fully gated unit is represented as follows:

Initially, for t = 0, the output vector is h_0 = 0

$$\boldsymbol{z_t = \sigma_g\ (W_z x_t + U_z h_{t-1} + b_z)} \tag{6- 7}$$

$$r_t = \sigma_g\ (W_r x_t + U_r h_{t-1} + b_r) \tag{6- 8}$$

$$\boldsymbol{h_t = z_t\ h_{t-1}} + (1 - \boldsymbol{z_t}) \otimes \emptyset\, \boldsymbol{h}\ (\boldsymbol{W_h x_t} + \boldsymbol{U_h}\ (\boldsymbol{r_t} \otimes \boldsymbol{h_{t-1}}\ ) + \boldsymbol{b_h}) \tag{6- 9}$$

Were ⊗ is the Hadamard product. W, U, b are parameter matrices and vectors. σ_g and $\emptyset\, h$ are the activation functions, σ_g is a sigmoid function and ∅ h a hyperbolic tangent

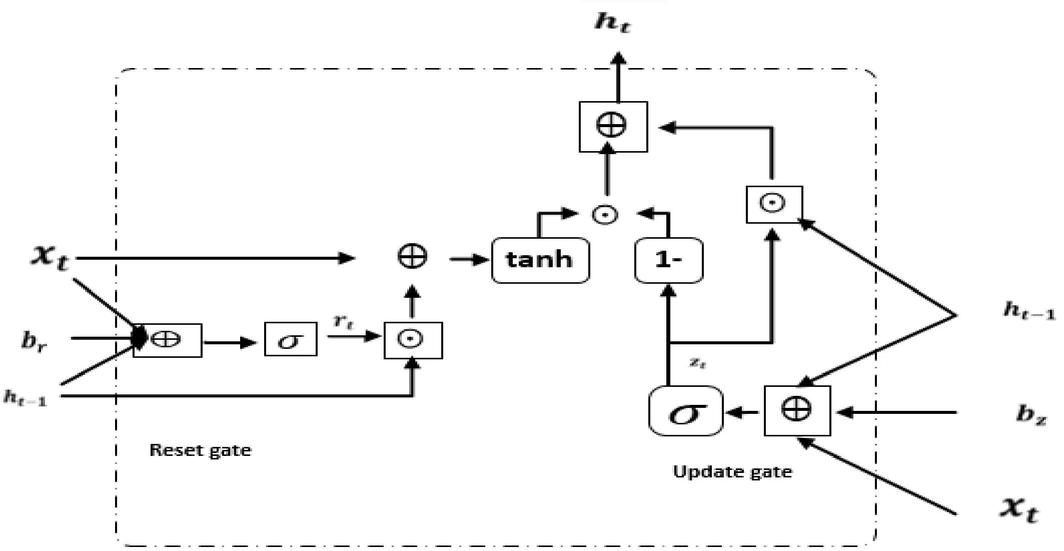


Figure 6- 3 A GRU block with an update and reset gate, sigmoid and hyperbolic tangent

The BGRU section of the model is designed as follows. First, the BGRU cells are constructed so that the result of feedforward is computed (F_t) and the feedback propagation (B_t) are merged at the first BGRU layer. Four different methods can merge the outcome, concatenation (default), summation, multiplication, and average. In this study, we will compare the performance of each merging method. The merging is represented as follows

$$\boldsymbol{O_t^1} = \boldsymbol{concat}\ ((\overrightarrow{\boldsymbol{F_t}}), (\overleftarrow{\boldsymbol{B_t}})) \tag{6- 10}$$

Such that$(\overrightarrow{F_t}) = (\ \overrightarrow{h_1}, \overrightarrow{h_2}, \overrightarrow{h_3}, \dots, \overrightarrow{h_t})$

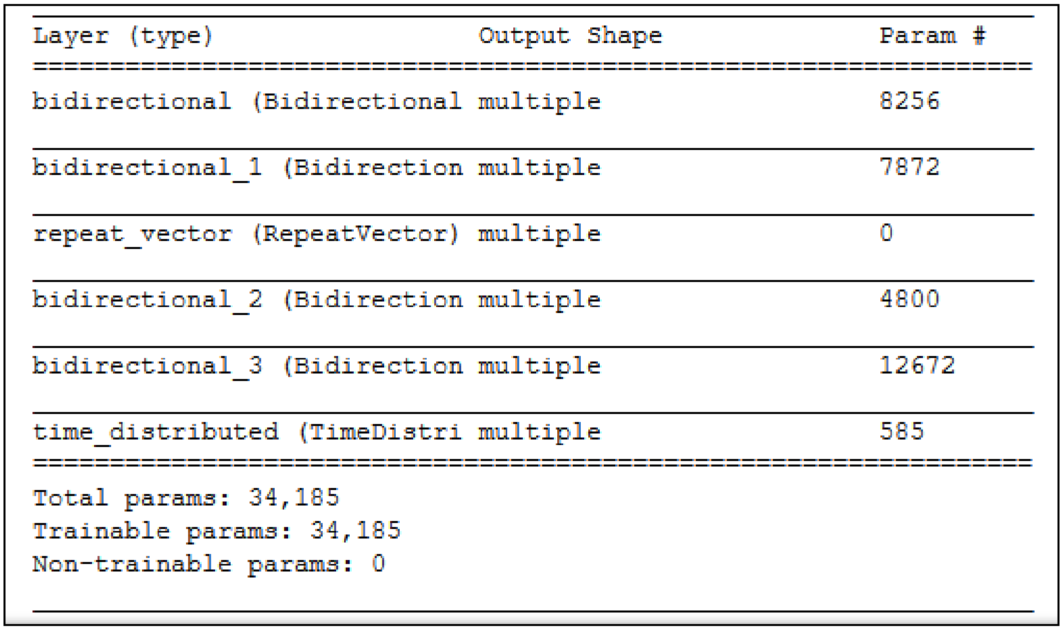
And $(\overleftarrow{B_t}) = (\ \overleftarrow{h_t}, \overleftarrow{h_{t+1}}, \overleftarrow{h_{t+2}}, \overleftarrow{h_{t+3}}, \dots \overleftarrow{h_n},)$

Second, a fully connected layer is used to multiply the BGRU network's output with its weight and bias. Then a SoftMax regression layer makes a prediction using input from the fully connected layer. A weighted classification layer is used to compute the weighted cross-entropy loss function for prediction score and training target, which helps tackle the imbalanced classification problem. The following loss is used

$$(\boldsymbol{p_{,t}}) = -\ (1 - (\boldsymbol{p_t}))\ ^{\boldsymbol{\gamma}}\boldsymbol{log_2}\ (\boldsymbol{p_t}) * \boldsymbol{\theta_i} \tag{6- 11}$$

Where $(p_{,t})$ represent the estimated probability of each class, and $\gamma \geq 0$ is the discount factor parameter that can be tuned for best estimation and θ_i is the logic weight of each class

Table 6- 1 Proposed BGRU Architecture



```
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional (Bidirectional multiple                  8256

bidirectional_1 (Bidirection multiple                  7872

repeat_vector (RepeatVector) multiple                  0

bidirectional_2 (Bidirection multiple                  4800

bidirectional_3 (Bidirection multiple                  12672

time_distributed (TimeDistri multiple                  585
=================================================================
Total params: 34,185
Trainable params: 34,185
Non-trainable params: 0

```

6.1.3.3 The convolutional neural networks

The use of deep learning approaches to process time-series data has recently been shown to produce improved results [53]. One of the deep learning approaches that have been widely used is convolutional neural networks (CNN). CNN's popularity is attributed to its capability to read, process, and extract the most important features of two-dimensional data, contributing to its performance improvement, especially for image classification [54][55]. Such data can be transformed to suit CNN in a scenario where the input data are not images [56]. Time series data is one of those data structures that can be transformed for CNN applications. Figure 6-4 shows a time-series dataset of length M and width N, where the length is the number of timesteps in the data, and the width is the number of variables in a multivariate time series. In transforming the times series data for CNN [57][58], a 1D convolutional kernel would be of the same width (number of variables). The kernel will then move top to down performing convolutions until the end of the series. The time series elements covered at a given time (window) are multiplied by the convolutional kernel elements. The multiplication result is added, and a non-linear activation function is applied to the value. The resulting value becomes an element of the next new filtered series. The kernel then moves forward to produce the next value. Max-pooling is applied to each of the filtered series of vectors. The vector's largest value is chosen, which is used as an input to a regular, fully connected layer.

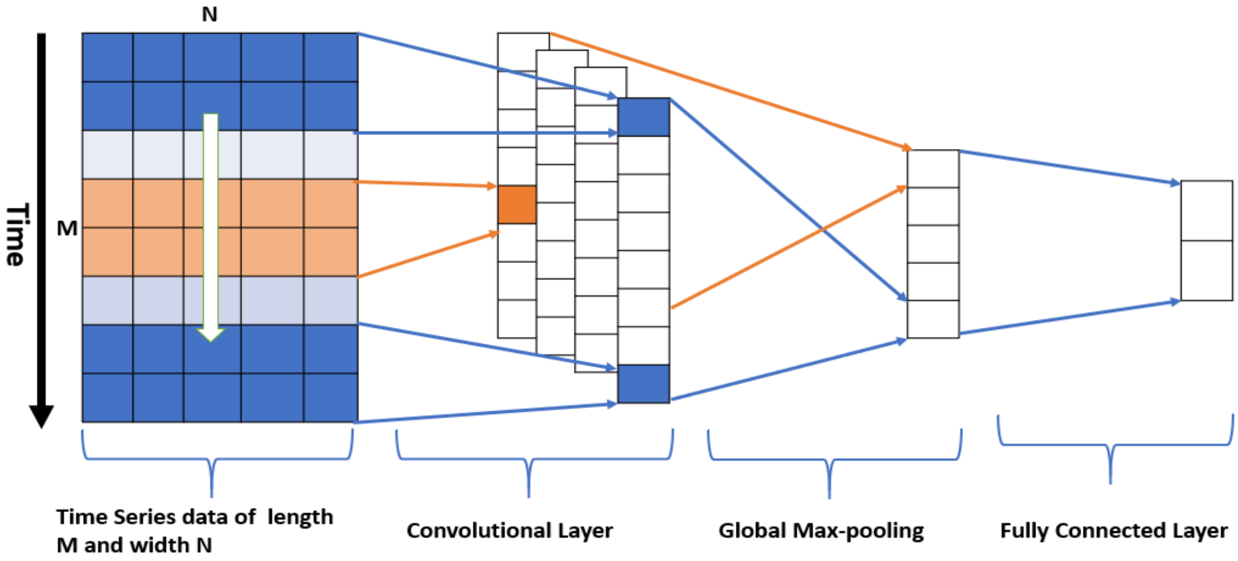


Figure 6- 4 CNN structure for Time Series Data

In designing the structure of BGRU with CNN, there is no out of the box or defined rule of thumb approach. Standard artificial neural network structure usually consists of an input layer, one or more hidden layers, and an output layer. To obtain an optimal result, the number of hidden layers and neurons used depends on the individual problem, and it is often a trial and error process. The most common approach is the use of K-fold cross-validation, as seen in [59]–[61]. However, for evaluation, some k number of nodes need to be defined, which can be obtained by a simple formula,

$$\boldsymbol{M_k} = \frac{\boldsymbol{M_s}}{\boldsymbol{\alpha(M_i+M_0)}} \tag{6- 12}$$

where M_s is the total number of samples in the training data, $M_i\ and\ M_0$ are a number of input and output neurons, respectively, and α is the scaling factor. For example, if α is set between two to ten, it means we can calculate eight different numbers to feed into the validation process to obtain an optimal result. The number of parameters to train is computed as equation 6-5 to 6-11, the number of inputs in the first layer equals the defined window size, and the number of folds to use in the cross-validation. The subsequent layers have a number of outputs of the previous layer as input. A simulation is conducted, and the training and testing error is plotted over the number of neurons in the hidden layer. The number of neurons is chosen that minimises the test error while keeping an eye on overfitting. Because the problem is formulated as binary classification and the data is extremely imbalanced, we use a modified loss function (equation 6-10), and SoftMax as the final activation function.

6.1.3.4 Proposed Method

Our objective is to develop a model that will detect and also predict rare extreme failure from the large log-based dataset. As seen in Figure 6-5, the basic idea is to separate the prediction of rare failure from its detection. Therefore, the proposed model employs two stages, detecting rare failure using auto-encoder and predicting the next occurrences of that failure using BGRU and CNN architectures.

The choice of the BGRU in the design is to capture a long dependency bidirectionally (forward and backwards) to enable effective learning. The rationale behind the choice of method is based on the nature of the dataset (i.e. heterogeneous and time series in nature). Usually, time-series datasets are mainly trained using recurrent neural networks (RNN); the challenge with RNN's is that they suffer from vanishing gradient problems and has a short-term memory. Varnishing gradient problem arises when training a deep multi-layer RNN (feedforward network) with a gradient-based learning approach and back-propagation. In the process, the weight of each ANN is updated in proportion to the partial derivatives of the error function with respect to weight in each iteration. The problem arises when useful gradient information is unable to propagate from the out layer back to the input layer of the model. In order to solve the vernishing gradient problem in RNN, the gated recurrent unit (GRU) networks were developed to capture long time dependencies in the sequence learning and to handle the gradient vanishing problem through the use of modified hidden layers or gates.

Convolutional Neural Network (CNN) uses a process known as convolution when determining a relationship between available variables in the dataset [20]. For example, in convolutional learning, given two functions f and g, the convolution integral expresses how the shape of one function is modified by the other. Traditionally, CNN's were designed to process multi-dimensional data, such as image classification, not to account for sequential dependencies like in RNNs, LSTMs or GRUs [62]. Therefore, The key benefit of adding CNN layers for sequential learning is its ability to use filters bank [63] to compute dilations between each cell, also referred to as "dilated convolution", which in turn allows the network layers in CNN to understand better the relationships between the different variables in the dataset, generating improved results.

The dataset is extremely imbalanced; that is, the imbalanced ratio between the positively labelled and negatively labelled data is less than 5% of the total. In such an extremely rare problem, traditional deep learning algorithms are overwhelmed by the majority class, producing bias result in detriment to the minority class [42], [64]. Therefore, we proposed AE-CNN-BGRU to handle the problem differently. The framework of the proposed model is shown in Figure 6-5. At the detection stage, the first AE model is used to detect rare failures using reconstruction errors. The data is divided into positive labelled (rare minority class) and negatively labelled (majority class). The AE model is then trained with only negatively labelled data (X_{-ve}) by feeding the encoder layer of AE with the original negatively labelled data. The latent code, which represents a compresses feature, is extracted in the middle layer. The decoder layers will then reconstruct the original data using compressed latent code as input. After the encode-decode process, a reconstruction error is known, which also shows the highest error that is later used for threshold setting. Since the AE model is first trained using negatively labelled data when the data is combined (X_t) and fed into the AE model. An anomaly can easily be detected because any data point coming from the negatively labelled class is expected to have a low error, and if coming from a positive class, the error will be higher. The low error is because it is coming from the same data used to train the first section AE model (as seen in the detection phase of Figure 6-5). On the other hand, when a new data point is from a positively labelled class, it is expected to have a higher reconstruction error score, which will be an anomaly.

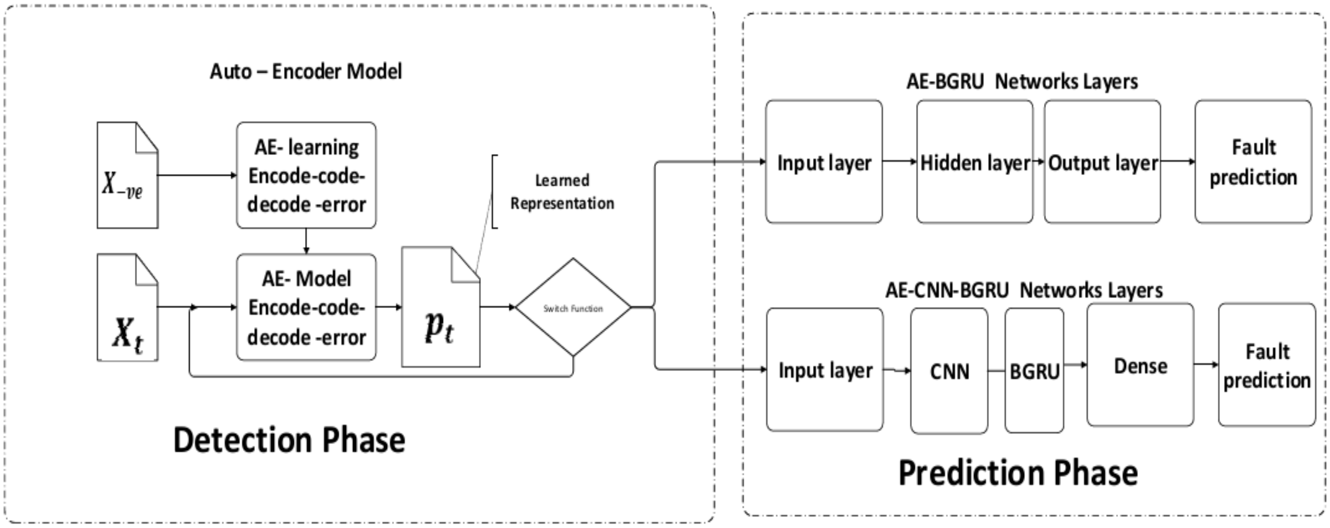


Figure 6- 5 An integrated AE, BGRU, and CNN networks for rare fault detection and prediction

For example, when a datapoint x_t is fed into the AE model, it will be classified as a fault if the reconstruction error exceeds a defined threshold; otherwise, it will be classified as no-fault. Once the faults are identified, the resulting compressed data is then fed into the next section of the framework, which is the AE-BGRU or AE-CNN-BGRU model for the failure prediction. The input data to the prediction model is the learned latent representation of the original dataset. To determine a threshold that offers the best result. We construct a function that iterates through a loop using precision and recall until the desired threshold is obtained.

6.1.4 Case Study and Experimental Setup

The fundamental research question of interest is whether AE-BGRU or AE-CNN-BGRU, with explicit failure detection and additional training capability, can outperform the normal unidirectional deep learning time-series methods on an extremely imbalanced dataset. Another important question is, can model performance for rare failure prediction be improved if learning is done in two directions (feedforward and feedback propagation). Also, how different does the architecture of deep learning models treat the input data? We conduct a series of experiments to investigate the above questions and report the result. The experiment is set up to verify the performance of our proposed approach in handling the rare occurrences of failure. Therefore, we use the log-based aircraft central maintenance system data, which comprises aircraft failure and warning messages. The following experiment was conducted.

1. To investigate whether the proposed AE-BGRU model has a performance advantage over the normal GRU model in predicting rare aircraft component failure.

2. To investigate if additional layers of training in the AE-CNN-BGRU model architecture can improve model performance.

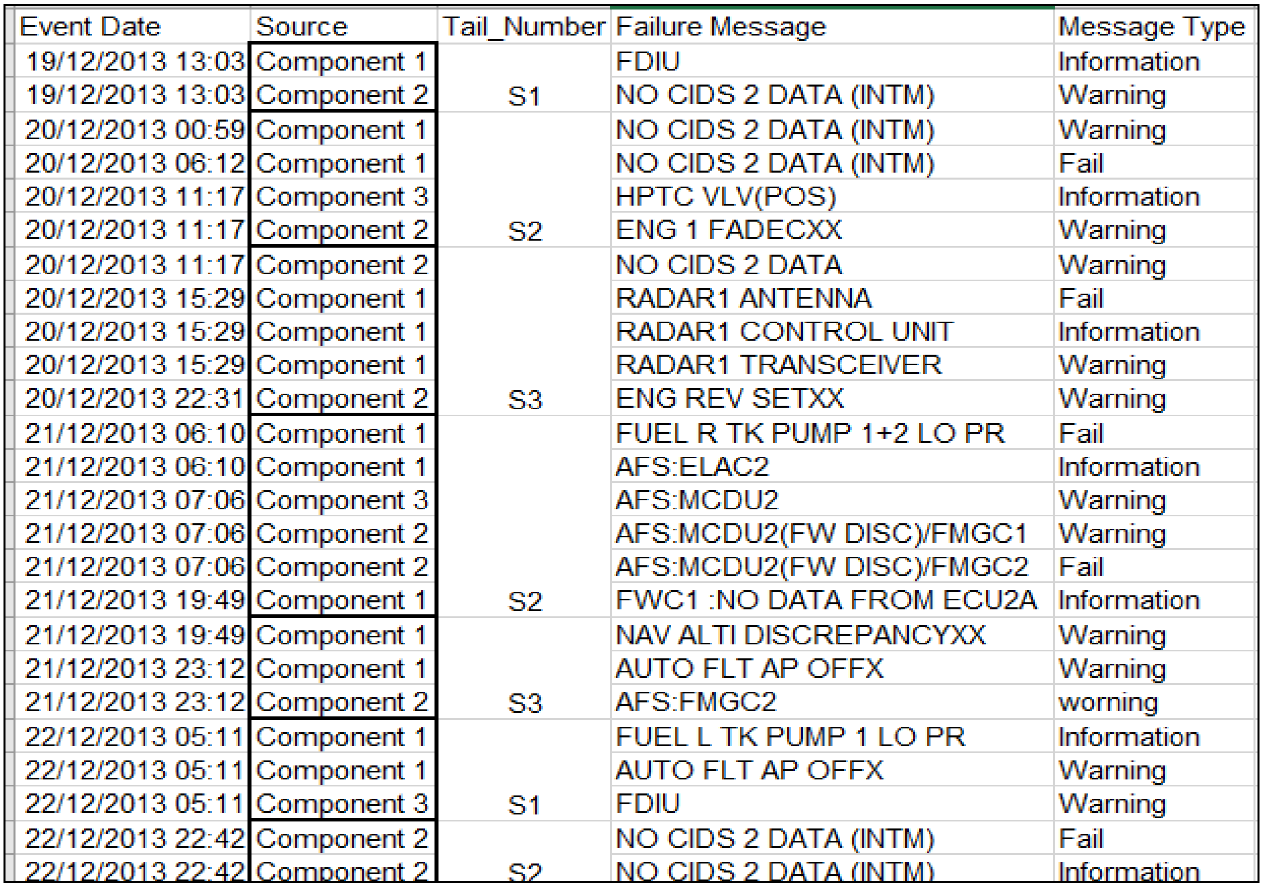
3. To investigate if training the proposed model using an extremely imbalanced dataset in a bidirectional way (forward and backwards) can improve model performance.

4. To provide a deep learning architecture performance analysis for the rare failure prediction via the Log-based ACMS dataset.

We categorise the modelling approach into two, binary class and multi-class. In the first scenario, we modelled it as a multi-class classification problem that predicts all the targeted component failures at the same time. Secondly, we modelled it as a binary classification problem that is predicting individual functional item

6.1.4.1 Dataset

This study uses over eight years' worth of data recorded from more than 60 aircraft. The dataset is collected from two databases. The first database is the Aircraft Central Maintenance System (ACMS) data, which comprises error messages from BIT (built-in test) equipment (that is, aircraft fault report records) and the flight deck effects (FDE). These messages are generated at different stages of flight phases (take-off, cruise, and landing). The second database is the record of aircraft maintenance activities (i.e. the comprehensive description of all recorded aircraft maintenance activities). The dataset is obtained from a fleet comprised of A330 and A320 aircraft. Some components are identified by functional item Number (FIN) chosen for validation. The target components are chosen based on their high practical value and an adequate number of known failure cases. The other consideration for the choice of the component is those that are replaced due to unscheduled. Figure 6-6 shows an example of the ACMS dataset.



Event Date	Source	Tail_Number	Failure Message	Message Type
19/12/2013 13:03	Component 1	S1	FDIU	Information
19/12/2013 13:03	Component 2		NO CIDS 2 DATA (INTM)	Warning
20/12/2013 00:59	Component 1	S2	NO CIDS 2 DATA (INTM)	Warning
20/12/2013 06:12	Component 1		NO CIDS 2 DATA (INTM)	Fail
20/12/2013 11:17	Component 3		HPTC VLV(POS)	Information
20/12/2013 11:17	Component 2		ENG 1 FADECXX	Warning
20/12/2013 11:17	Component 2	S3	NO CIDS 2 DATA	Warning
20/12/2013 15:29	Component 1		RADAR1 ANTENNA	Fail
20/12/2013 15:29	Component 1		RADAR1 CONTROL UNIT	Information
20/12/2013 15:29	Component 1		RADAR1 TRANSCEIVER	Warning
20/12/2013 22:31	Component 2		ENG REV SETXX	Warning
21/12/2013 06:10	Component 1	S2	FUEL R TK PUMP 1+2 LO PR	Fail
21/12/2013 06:10	Component 1		AFS:ELAC2	Information
21/12/2013 07:06	Component 3		AFS:MCDU2	Warning
21/12/2013 07:06	Component 2		AFS:MCDU2(FW DISC)/FMGC1	Warning
21/12/2013 07:06	Component 2		AFS:MCDU2(FW DISC)/FMGC2	Fail
21/12/2013 19:49	Component 1		FWC1 :NO DATA FROM ECU2A	Information
21/12/2013 19:49	Component 1	S3	NAV ALTI DISCREPANCYXX	Warning
21/12/2013 23:12	Component 1		AUTO FLT AP OFFX	Warning
21/12/2013 23:12	Component 2		AFS:FMGC2	worning
22/12/2013 05:11	Component 1	S1	FUEL L TK PUMP 1 LO PR	Information
22/12/2013 05:11	Component 1		AUTO FLT AP OFFX	Warning
22/12/2013 05:11	Component 3		FDIU	Warning
22/12/2013 22:42	Component 2	S2	NO CIDS 2 DATA (INTM)	Fail
22/12/2013 22:42	Component 2		NO CIDS 2 DATA (INTM)	Information

Figure 6- 6 Example of the real ACMS dataset. Sensitive data elements have been masked

Data from the year 2011 to 2016 is used for training, while the remaining data from 2016 to 2018 is used for testing. The targeted LRU's from the A330 aircraft family are **4000KS** - Electronic Control Unit/ Electronic Engine Unit, **4000HA** – Pressure Bleed Valve, and **438HC** – Trim Air Valve. From A320 are **11HB** – Flow control valve, **10HQ** - Avionics equipment ventilation computer, **1TX1** - Air traffic service unit.

6.1.4.2 Sensitivity analysis for BGRU Merge Modes

Sensitivity analysis was carried out to determine the best merging mode that can be used to integrate the outcomes of the BGRU layers for the proposed model. As shown in Figure 6-7, plotting loss against epoch, the line plot is created to compare the four merge modes (summation, concatenation, multiplication and average). A time-series data of size 10000 was generated and trained, using a loss shown in equation (6-1) and running the BGRU networks for 200 epochs. The result indicates that concatenation (the green line) is the best merge mode because it has lower loss values.

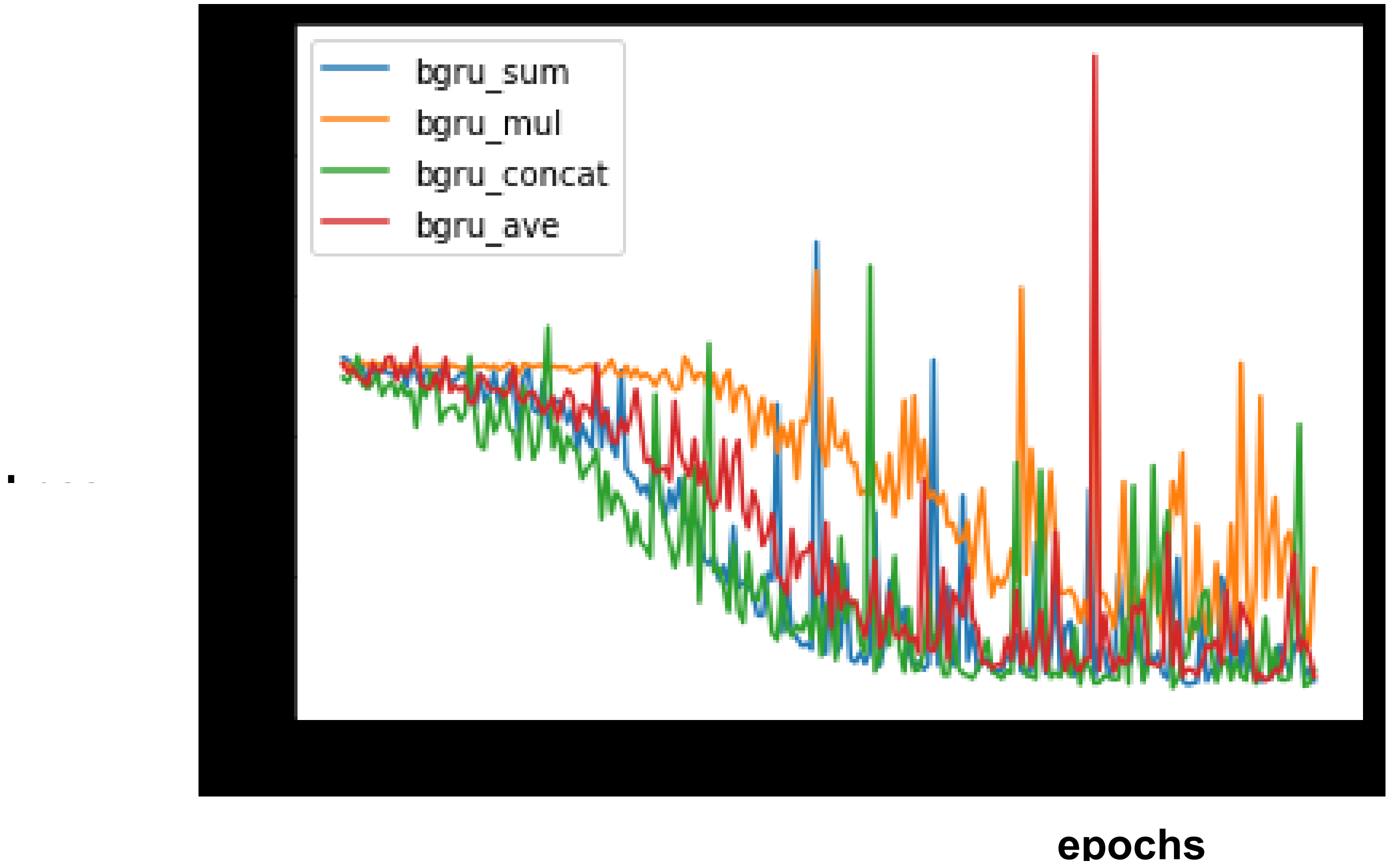


Figure 6- 7 Comparing BGRU Merge Modes. The figure shows the analysis to determine the merging mode that can be used for the BGRU layers in the proposed AE-CNN-BGRU model. The target is to choose the best merging method (i.e. with lower error).

Further analysis was carried out to determine the effect of bidirectional networks as compared to unidirectional ones. Three network architectures were set up for the analysis, two unidirectional (the forward and the backwards networks) and the bidirectional network. The result is shown in Figure 6-8; as observed, the GRU forward and GRU backwards shows a similar pattern, while BGRU_concat (green) shows a better loss (low errors). The comparison result indicates that BGRU can add performance improvement, not just merely reversing the input sequence.

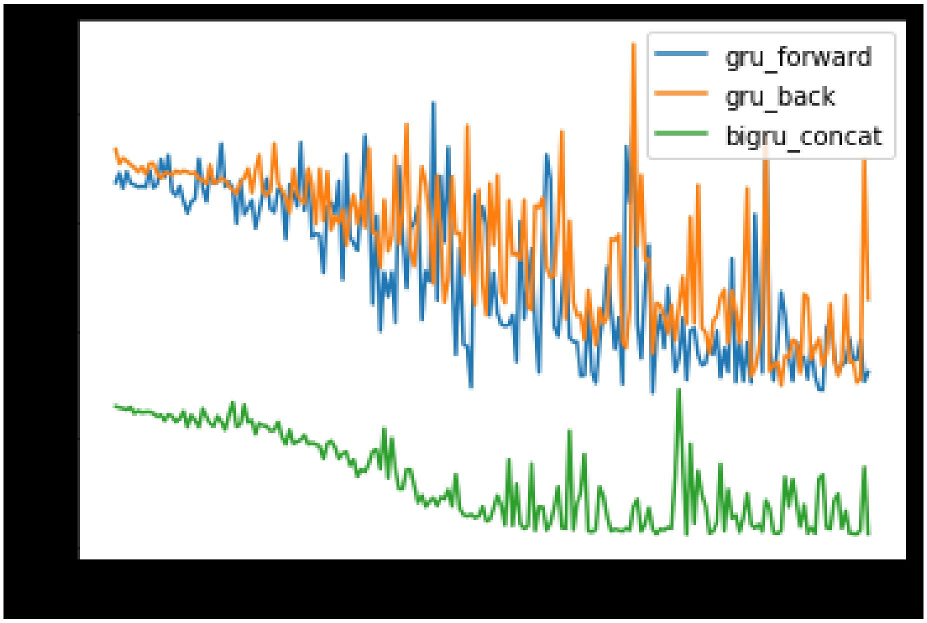


Figure 6- 8 Comparing GRU with BGRU

6.1.5 Result and Discussion

A study is conducted to determine if training the model using an imbalanced dataset, using bidirectional models, can improve the minority class's detection. Two bidirectional models were considered, the AE-BGRU and the AE-CNN-BGRU models, and compared with GRU (baseline), the result is shown in Table 6-1. As before, the models are validated using data from two families of aircraft (A330 and A320); in the A330 and A320 aircraft family, the size of the training dataset is 360575 389829 respectively. The target is to predict the replacement of aircraft LRU identified by their functional identification numbers (FINs). The validation result is based on the validation (testing) data, and the size is dependent on the number of patterns related to each target component. The targeted number of failure for each component are 4000KS =11, 4000HA =13, 438HC = 9, 11HB=6, 10HQ =8 and 1TX1 =15.

As observed, the proposed models show superior performance compared to baseline. Considering the A330 dataset and training the proposed algorithms to predict each component's failure, it can be observed that after validation. The result for predicting failure of 4000KS (the aircraft electronic engine unit) using the AE-BGRU model records a precision of 72%, recall of 61%, g-mean 67%, and a false-positive rate of 0.091%. AE-CNN-BGRU model achieves a precision of 90%, recall of

66%, g-mean of 77%, and a false positive rate of 0.011%. Compared to normal GRU with a precision of 60%, recall 0.55%, g-mean 53%, and a false-positive rate of 0.005. A similar result is seen for the other components, the 4000HA (pressure bleed valve) and the 438HC (trim air valve).

When using data from the A320 aircraft family, the results also indicate superior performance for the proposed AE-BGRU and AE-CNN-BGRU models as compared to unidirectional GRU. The result for predicting the failure of 11HB (Flow control valve) indicates that AE-CNN-BGRU achieved a precision of 66%, recall 59%, g-mean 67%, and a false-positive rate of 0.019% compared to GRU with a precision of 61%, recall 51% g-mean 49% and false positive rate of 0.005. Similar performance is seen for other components, the 10HQ - Avionics equipment ventilation computer and 1TX1 - Air traffic service unit.

Table 6- 2 Aircraft A330 and A320 rare failure prediction of individual LRU's using ACMS dataset

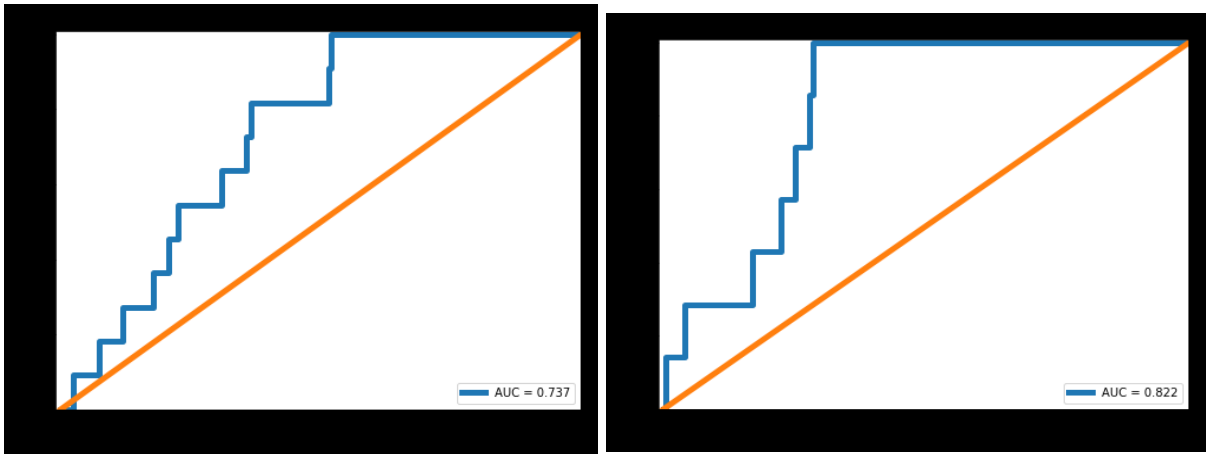
Aircraft ACMS Dataset													
			GRU (Baseline)				**AE-BGRU**				**AE-CNN-BGRU**		
	LRU's	**IR**	**P**	**R**	**GM**	**FPR**	**P**	**R**	**GM**	**FNR**	**P**	**R**	**GM**	**FNR**
A330-Family	**4000KS**	**0.0043**	**0.60**	**0.55**	**0.53**	**0.005**	**0.720**	**0.61**	**0.67**	**0.00091**	**0.909**	**0.66**	**0.778**	**0.00011**
	4000HA	**0.0047**	**0.41**	**0.40**	**0.41**	**0.008**	**0.538**	**0.538**	**0.632**	**0.00127**	**0.769**	**0.768**	**0.769**	**0.000638**
	438HC	**0.0044**	**0.54**	**0.51**	**0.53**	**0.006**	**0.666**	**0.600**	**0.632**	**0.00083**	**0.88**	**0.610**	**0.730**	**0.00027**
A320 Family	**11HB**	**0.0028**	**0.62**	**0.51**	**0.49**	**0.005**	**0.660**	**0.58**	**0.624**	**0.00019**	**0.66**	**0.59**	**0.671**	**0.00019**
	10HQ	**0.0031**	**0.60**	**0.51**	**0.55**	**0.006**	**0.625**	**0.49**	**0.55**	**0.00028**	**0.75**	**0.66**	**0.707**	**0.000191**
	1TX1	**0.0064**	**0.66**	**0.52**	**0.58**	**0.007**	**0.866**	**0.764**	**0.814**	**0.00029**	**0.85**	**0.741**	**0.860**	**0.000193**

**** *LRU's* represents an aircraft line replacement unit. *P* is precision, *R* is recall, *GM* is g-mean, *FPR* is a false positive rate.**

In the six FINs considered, the proposed models significantly improve the false-positive rate, which is very important for any predictive maintenance model acceptability. Also, the AE-CNN-BGRU model shows an overall improvement of 25% in precision, 14% in recall, and 2% in G-mean.

6.1.5.1 Measuring the Success Rate of the Proposed Models Using A330 Aircraft

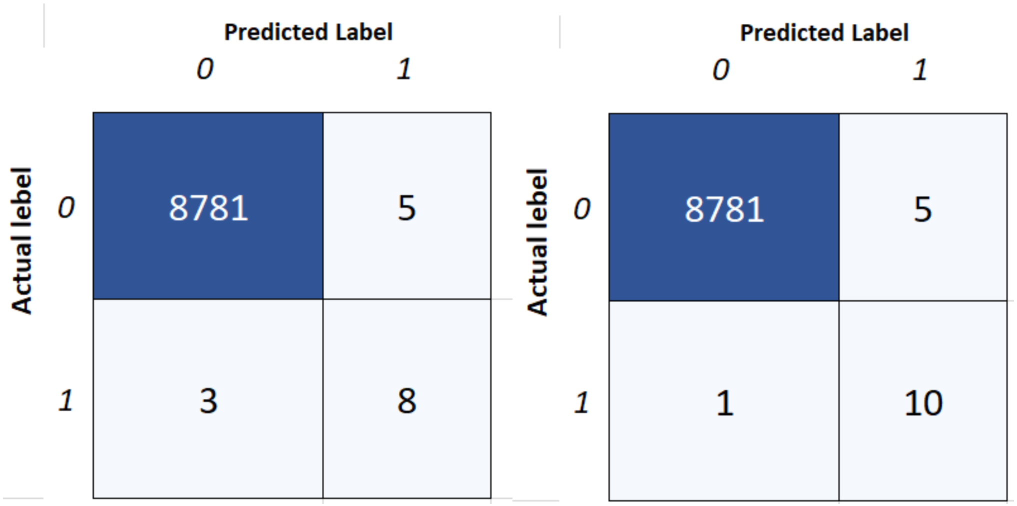
Figure 6-9 shows the ROC curve for the proposed models AE-BGRU and the AE-CNN-BGRU. The ROC curve for the AE-CNN-BGRU model (Figure 6-9 (b)) shows AUC= 0.822 indicates that there is an 82.2% chance that the model will be able to distinguish between positive classes (component failure) and negative class (non-failure). In contrast, Figure 6-9 (a) shows the ROC curve for the AE-BGRU model with AUC = 0.737, which indicates that the model has a 73.7% chance of distinguishing between classes.



(a) **(b)**

Figure 6- 9 ROC curve for FIN_4000KS prediction using (a) AE-BGRU and (b) AE-CNN-BGRU models

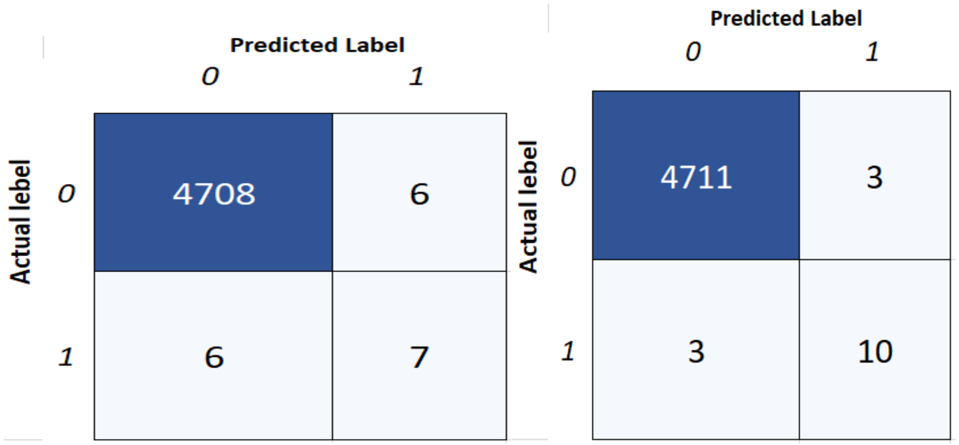
Also, to measure the model success rate in predicting extremely rare failure, a confusion matrix was plotted for both proposed models. Figure 6-10 shows a confusion matrix for predicting the failure of the electronic engine unit (FIN_4000KS). Figure 6-10(a) AE-BGRU model predicted eight failures correctly out of the eleven true failures, and Figure 6-10(b) shows that the AE-CNN-BGRU model predicted ten out of eleven. This prediction includes 10 flight legs in advance. It can also be observed that the AE-CNN-BGRU model predicts approximately 94% of extremely rare failure of components, which is a reasonable specificity, especially for aircraft maintenance acceptability.



(a) **(b)**

Figure 6- 10 Confusion matrix for FIN_4000KS using (a) AE-BGRU and (b)AE-CNN-BGRU model

As seen in Figure 6-11(a), AE-BGRU predicted 7 out of 13 and Figure 6-11(b) AE-CNN-BGRU 10 out of 13 unplanned replacement of pressure bleed valve (FIN_4000HA) failures. This prediction includes 10 flight legs in advance, and it can also be observed that the AE-CNN-BGRU model shows superior performance. A similar performance is observed for other components tested. The general result indicated that the proposed AE-CNN-BGRU model detected and predicted approximately 80% of extremely rare failures, which is a reasonable specificity, especially for aircraft maintenance.

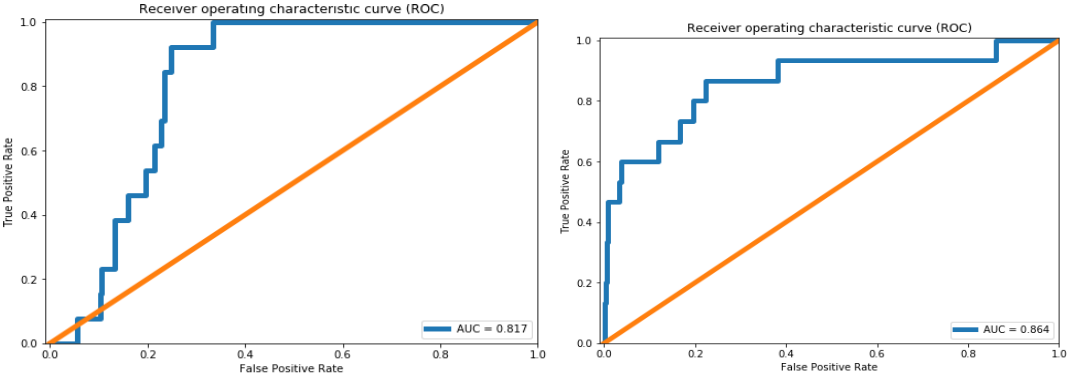


(a) (b)

Figure 6- 11 Confusion matrix for FIN_4000HA using AE-BGRU and AE-CNN-BGRU model

6.1.5.2 Measuring the success rate of the proposed models using A320 Aircraft

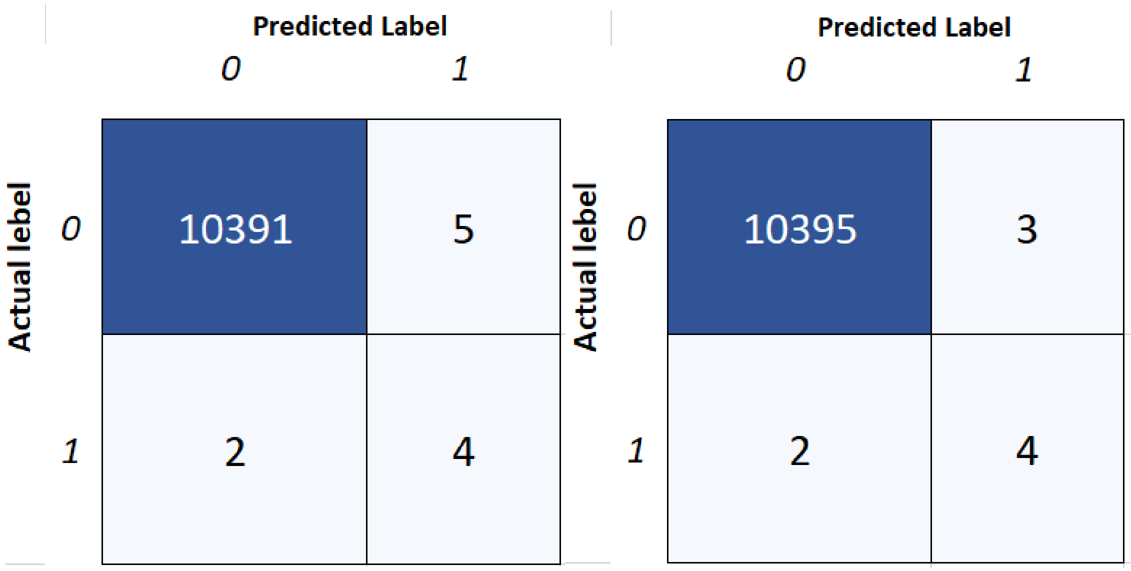
Figure 6-12 shows the ROC curve for the proposed models AE-BGRU and the AE-CNN-BGRU. The ROC curve for the AE-CNN-BGRU model (Figure 6-12 (b)) shows AUC= 0.864, which indicates that there is an 86.4% probability that the model will be able to distinguish between positive class (component failure) and negative class (non-failure). In contrast, Figure 6-12 (a) shows the ROC curve for the AE-BGRU model with AUC = 0.817, which indicates that the model has an 81.7% probability of distinguishing between classes. The result indicated that AE-CNN-BGRU has an 8% better classification performance compared to AE-BGRU



(a) (b)

Figure 6- 12 ROC curve for predicting 11HB using (a)AE-BGRU and (b) AE-CNN-BGRU

As seen in Figure 6-13(a), AE-BGRU predicted 4 out of 6 and Figure 6-13 (b) AE-CNN-BGRU 4 out of 6 unplanned replacement of pressure bleed valve (FIN_11HB). This prediction includes 10 flight legs in advance. A similar performance is observed for other components tested. The general result indicated that the proposed AE-CNN-BGRU model detected and predicts approximately 50% of extremely rare failures.



(a) (b)

Figure 6- 13 Confusion matrix for FIN_11HB using AE-BGRU and AE-CNN-BGRU model

Although both models predicted 50% of the failure, it can be observed that the AE-CNN-BGRU model shows superior performance in terms of recall. A good recall indicates that the model has a good potential measure of correctly identifying true positives.

6.1.5.3 Sensitivity of AE-CNN-BGRU model to design parameters

Additional analysis was carried out to determine if adding CNN layers to the AE-BGRU network could improve performance. After the implantation, the result indicated that there was performance improvement as shown in Table 6-2, Figures 6-9 to 6-3 . The AE-CNN- BGRU model performance improvement can be accounted to the following factors. First, in training time-series dataset, especially using BGRU or LSTM. Such networks account for the sequential dependency in a situation where a correlation exists between the variables in the given dataset (a process known as autocorrelation); during training, a normal GRU/LSTM network would treat all the variables as independent, excluding any relationship that exist between both observed and latent variables. Whereas CNN uses a process known as convolution when determining a relationship between available variables in the dataset [20]. For example, in convolutional learning, given two functions f and g, the convolution integral expresses how the shape of one function is modified by the other.

Traditionally, CNN's were designed to process multi-dimensional data, such as in image classification, not to account for sequential dependencies like in RNNs, LSTMs or GRUs [62]. Therefore, The key benefit of adding CNN layers for sequential learning is its ability to use filters bank [63] to compute dilations between each cell, also referred to as "dilated convolution", which in turn allows the network layers in CNN to understand better the relationships between the different variables in the dataset, generating improved results.

6.1.5.4 Sensitivity of the models to the imbalanced ratio

A sensitivity analysis was carried out for the imbalanced ratio on the designed network architecture and the input data. As observed in Table 6-2 the six case considered have different imbalanced ration (400KS=0.0043, 4000HA=0.0047,438HC = 0.0044, 11HB =0.0028, 10HQ = 0.0031, 1TX1 = 0.0064). The components differed not only in the imbalanced ratio but also in distributions and failure patterns. As seen in Figure 6-14, It can be observed that the novel model (AE-CNN-GRBU) show a significant reduction in the false-negative rate as compared to others, indicating it is robust to different conditions of the dataset. Also, it is observed that the imbalance ratio impacts the false-negative rate for the test components from the A330 aircraft family (4000KS - Electronic Control Unit/ Electronic Engine Unit, 4000HA – Pressure Bleed Valve, and 438HC – Trim Air Valve). For example, 4000HA with the highest imbalance ratio of 0.0047 has a false negative rate of about 0.000639 compared to 4000KS with the lowest imbalanced ratio and false-negative rate of 0.00011. the analysis for A320 (11HB – Flow control valve, 10HQ - Avionics equipment ventilation computer, 1TX1 - Air traffic service unit) show insignificant changes to the imbalance ratio in terms of false-negative rate.

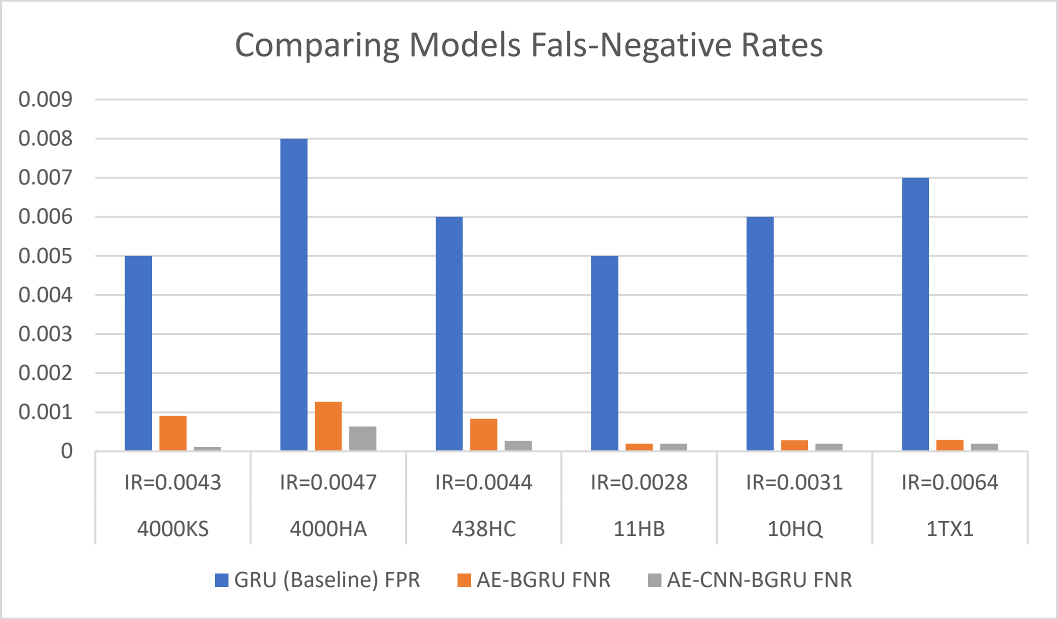


Figure 6- 14 Sensitivity analysis of Imbalanced ration against False Negative Rate

6.1.6 Conclusion and Future Work

This paper proposes a novel technique that can narrow down the volume of logs of aircraft warning and failure messages recorded by the central maintenance system into a small set of important and most relevant logs. The reduced log is then used to develop a model for aircraft' predictive maintenance, focusing on extremely rare failure predictions. The proposed model integrates an auto-encoder with bidirectional gated recurrent networks, which complement each other to generate accurate link failure/warning messages related to aircraft LRU removal and help identify irregular patterns and trends. The auto-encoder performs the detection of the rare failures, while the BGRU networks (with CNN) perform the prediction. The proposed technique is evaluated using real-world aircraft central maintenance system (ACMS) data. The evaluation results indicate that the AE-CNN-BGRU model can effectively handle irregular patterns and trends, mitigating the imbalanced classification problem. Comparing AE-CNN-BGRU with other similar deep learning methods, the proposed approach shows superior performance with 25% better precision, 14% in the recall, and 3% in g-mean. The results also indicate the model effectiveness in predicting component failure within a defined useful period that aids in minimising operational disruption. The superior performance indicates that the AE-CNN-BGRU model networks are able to capture the underlying temporal structure better by traversing the input data in a bidirectional manner (feedforward and feedback) while making the prediction. The performance improvement of AE-CNN-BGRU against the unidirectional GRU is understandable for certain types of data, such as in-text classification and prediction of text-to-words in sequence-to-sequence learning. However, it was not clear whether training extremely imbalanced, time-series data using a bidirectional approach would improve model performance as there may not be sufficiently definite temporal contexts and observable in-text sequence examples. Our results have clarified this question, showing that AE-CNN-BGRU outperforms normal GRU in the context of predicting rare failure in log-based aircraft ACMS datasets.

Further studies will be conducted on other architectures of AE-CNN-BGRU, such as transforming the time series into graphical representation using recurrence plots. The resulting images can be trained using CNN-BGRU for likely performance optimisation. Also, other aircraft data can be added to ACMS to enhance model training.

6.1.7 References

1. Dangut MD., Skaf Z., Jennions IK. Rare Failure Prediction Using an Integrated Auto-encoder and Bidirectional Gated Recurrent Unit Network. IFAC-PapersOnLine. Elsevier Ltd; 2020; 53(3): 276–282. Available at: DOI:10.1016/j.ifacol.2020.11.045

2. Kingsley-Jones M. Airbus sees big data delivering 'zero-AOG' goal within 10 years. Flightglobal. 2017; Available at: https://www.flightglobal.com/mro/airbus-sees-big-data-delivering-zero-aog-goal-within-10-years/126446.article

3. This D. Integration of smart maintenance and spare part logistics for healthcare systems Integration of Smart Maintenance and Spare Part Logistics for Healthcare Eindhoven University of Technology. 2018;

4. Krawczyk B. Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence. Springer Berlin Heidelberg; 2016; 5(4): 221–232. Available at: DOI:10.1007/s13748-016-0094-0

5. Dangut MD., Zakwan S., Jennions IK. Aircraft Predictive Maintenance Modeling using a Hybrid Imbalance Learning Approach. 2020. Available at: https://ssrn.com/abstract=3718065

6. Raghuwanshi BS., Shukla S. UnderBagging based reduced Kernelized weighted extreme learning machine for class imbalance learning. Engineering Applications of Artificial Intelligence. Elsevier Ltd; 2018; 74(July): 252–270. Available at: DOI:10.1016/j.engappai.2018.07.002

7. Wu Z., Lin W., Ji Y. An Integrated Ensemble Learning Model for Imbalanced Fault Diagnostics and Prognostics. IEEE Access. 2018; 6: 8394–8402. Available at: DOI:10.1109/ACCESS.2018.2807121

8. Wu Z., Guo Y., Lin W., Yu S., Ji Y. A weighted deep representation learning model for imbalanced fault diagnosis in cyber-physical systems. Sensors (Switzerland). 2018; 18(4). Available at: DOI:10.3390/s18041096

9. Nguyen KTP., Medjaher K. A new dynamic predictive maintenance framework using deep learning for failure prognostics. Reliability Engineering and System Safety. Elsevier Ltd; 2019;

188(March): 251–262. Available at: DOI:10.1016/j.ress.2019.03.018

10. Burnaev E. Rare Failure Prediction via Event Matching for Aerospace Applications. 2019; (July). Available at: http://arxiv.org/abs/1905.11586

11. Blancke O., Combette A., Amyot N., Komljenovic D., Lévesque M., Hudon C., et al. A Predictive Maintenance Approach for Complex Equipment Based on Petri Net Failure Mechanism Propagation Model. Proceedings of the European Conference of the PHM Society . 2018; 4(1): 1–12.

12. Blancke O., Komljenovic D., Tahan A., Combette A., Amyot N., Lévesque M., et al. A Predictive Maintenance Approach for Complex Equipment Based on Petri Net Failure Mechanism Propagation Model. Proceedings of the European Conference of the PHM Society . 2018; 1(4). Available at: https://www.researchgate.net/publication/326247314

13. Aivaliotis P., Georgoulias K., Arkouli Z., Makris S. Methodology for enabling digital twin using advanced physics-based modelling in predictive maintenance. Procedia CIRP. Elsevier BV; 2019; 81: 417–422. Available at: DOI:10.1016/j.procir.2019.03.072

14. Parris CJ. The Future for Industrial Services - The Digital Twin. Infosys Insights. 2016; : 42–49.

15. Okoh C., Roy R., Mehnen J. Predictive Maintenance Modelling for Through-Life Engineering Services. Procedia CIRP. The Author(s); 2017; 59(TESConf 2016): 196–201. Available at: DOI:10.1016/j.procir.2016.09.033

16. Phillips P., Diston D. A knowledge driven approach to aerospace condition monitoring. Knowledge-Based Systems. Elsevier BV; 2011; 24(6): 915–927. Available at: DOI:10.1016/j.knosys.2011.04.008

17. Ferri FAS., Rodrigues LR., Gomes JPP., De Medeiros IP., Galvao RKH., Nascimento CL. Combining PHM information and system architecture to support aircraft maintenance planning. SysCon 2013 - 7th Annual IEEE International Systems Conference, Proceedings. 2013; (April): 60–65. Available at: DOI:10.1109/SysCon.2013.6549859

18. Berberidis C., Angelis L., Vlahavas I. Inter-transaction association rules mining for rare events

prediction. Proc. 3rd Hellenic Conference …. 2004; Available at: http://lpis.csd.auth.gr/publications/076-Berberidis-Angelis-Vlahavas-SETN04.pdf

19. Salfner F., Lenk M., Malek M. A survey of online failure prediction methods. ACM Computing Surveys. 2010; 42(3). Available at: DOI:10.1145/1670679.1670680

20. Zhang K., Xu J., Min MR., Jiang G., Pelechrinis K., Zhang H. Automated IT system failure prediction: A deep learning approach. Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016. IEEE; 2016; : 1291–1300. Available at: DOI:10.1109/BigData.2016.7840733

21. Korvesis P., Besseau S., Vazirgiannis M. Predictive maintenance in aviation: Failure prediction from post-flight reports. Proceedings - IEEE 34th International Conference on Data Engineering, ICDE 2018. IEEE; 2018; : 1423–1434. Available at: DOI:10.1109/ICDE.2018.00160

22. Sipos R., Wang Z., Moerchen F. Log-based Predictive Maintenance. 2014; : 1867–1876.

23. Kallas M., Mourot G., Anani K., Ragot J., Maquin D. Fault detection and estimation using kernel principal component analysis. IFAC-PapersOnLine. 2017; 50(1): 1025–1030. Available at: DOI:10.1016/j.ifacol.2017.08.212

24. Lee J-M., Qin SJ., Lee I-B. Fault Detection of Non-Linear Processes Using Kernel Independent Component Analysis. The Canadian Journal of Chemical Engineering. 2008; 85(4): 526–536. Available at: DOI:10.1002/cjce.5450850414

25. Ismail Fawaz H., Forestier G., Weber J., Idoumghar L., Muller PA. Deep learning for time series classification: a review. Data Mining and Knowledge Discovery. 2019; 33(4): 917–963. Available at: DOI:10.1007/s10618-019-00619-1

26. Guo S., Yang T., Gao W., Zhang C. A novel fault diagnosis method for rotating machinery based on a convolutional neural network. Sensors (Switzerland). 2018; 18(5). Available at: DOI:10.3390/s18051429

27. Park P., Di Marco P., Shin H., Bang J. Fault detection and diagnosis using combined Autoencoder and long short-term memory network. Sensors (Switzerland). 2019; 19(21): 1–

17. Available at: DOI:10.3390/s19214612

28. Liu R., Yang B., Zio E., Chen X. Artificial intelligence for fault diagnosis of rotating machinery: A review. Mechanical Systems and Signal Processing. Elsevier Ltd; 2018; 108: 33–47. Available at: DOI:10.1016/j.ymssp.2018.02.016

29. Dangut MD., Skaf Z., Jennions IK. An integrated machine learning model for aircraft components rare failure prognostics with log-based dataset. ISA Transactions. Elsevier Ltd; 2020; (xxxx). Available at: DOI:10.1016/j.isatra.2020.05.001

30. Burnaev E. Rare Failure Prediction via Event Matching for Aerospace Applications. 2019 3rd International Conference on Circuits, System and Simulation, ICCSS 2019. 2019; : 214–220. Available at: DOI:10.1109/CIRSYSSIM.2019.8935598

31. Che C., Wang H., Fu Q., Ni X. Combining multiple deep learning algorithms for prognostic and health management of aircraft. Aerospace Science and Technology. Elsevier Masson SAS; 2019; 94: 105423. Available at: DOI:10.1016/j.ast.2019.105423

32. Baldi P. Autoencoders, Unsupervised Learning, and Deep Architectures. ICML Unsupervised and Transfer Learning. 2012; : 37–50. Available at: DOI:10.1561/2200000006

33. Le Q V. A Tutorial on Deep Learning Part 2: Autoencoders, Convolutional Neural Networks and Recurrent Neural Networks. Tutorial. 2015; : 1–20. Available at: DOI:https://cs.stanford.edu/~quocle/tutorial2.pdf

34. Farzad A., Gulliver TA. Log Message Anomaly Detection and Classification Using Auto-B/LSTM and Auto-GRU. 2019; : 1–28. Available at: http://arxiv.org/abs/1911.08744

35. Konar A. Artificial Intelligence and Soft Computing. Artificial Intelligence and Soft Computing. 1999. Available at: DOI:10.1201/9781420049138

36. Savoy J., Gaussier E. Information retrieval. Handbook of Natural Language Processing, Second Edition. 2010. 455–484 p. Available at: DOI:10.4324/9781351044677-24

37. Munna MTA., Alam MM., Allayear SM., Sarker K., Ara SJF. Prediction model for prevalence of type-2 diabetes complications with ANN approach combining with K-fold cross validation and K-means clustering. Lecture Notes in Networks and Systems. Springer International

Publishing; 2020. 1031–1045 p. Available at: DOI:10.1007/978-3-030-12388-8_71

38. Applications C. Mathematical and Computational Applications,. 2011; 16(3): 702–711.

39. Jiang P., Chen J. Displacement prediction of landslide based on generalised regression neural networks with K-fold cross-validation. Neurocomputing. Elsevier; 2016; 198: 40–47. Available at: DOI:10.1016/j.neucom.2015.08.118

40. Bengio Y., Courville A., Vincent P. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence. IEEE; 2013; 35(8): 1798–1828. Available at: DOI:10.1109/TPAMI.2013.50

41. Roc B. Comparing Two ROC Curves – Independent Groups Design. NCSS, LLC. 2021; : 1–26.

6.2 Rare Failure Prediction Using an Integrated Auto-encoder and Bidirectional Gated Recurrent Unit Network

This section is about a paper presented at a conference (IFAC-PapersOnLine 53.3 (2020): 276-282. DOI: 10.1016/j.ifacol.2020.11.045), which is expanded for journal publishing in section 6.1.

Aircraft fault detection and prediction is a critical element of preventing failures, reducing maintenance costs, and increasing fleet availability. This paper considers the problem of rare failure prediction in the context of aircraft' predictive maintenance. It presents a novel approach to predicting extremely rare failures, based on combining two deep learning techniques, auto-encoder (AE) and Bidirectional Gated Recurrent Unit (BGRU) network. AE is modified and trained to detect rare failure, and the result from AE is fed into the BGRU to predict the next occurrence of failure. The applicability of the proposed approach is evaluated using real-world test cases of log-based warning and failure messages obtained from the aircraft central maintenance system fleet database and the records of maintenance history. The proposed AE-BGRU model is compared with other similar deep learning methods. The proposed approach is 25% better in precision, 14% in recall, and 3% in G-mean. The result also shows robustness in predicting failure within a defined useful period.

6.2.1. Introduction

Fault detection, diagnosis, and prognosis (FDDP) have a huge potential to improve aircraft operational reliability and availability since the main aim of FDDP is to minimize losses while ensuring the safety of equipment and reducing the risk of unplanned breakdowns [1]. FDDP involves detecting the occurrence of fault as early as possible, classifying the fault type accurately, and predicting the next occurrences of such a fault. FDDP models are designed to detect anomalies of critical components by analyzing historical data to provide actionable alerts to the operators [2]. The operational and maintenance datasets generated in modern aircraft have become much larger as both the number of samples and the dimensionality have increased. Hence, implementing traditional physics-based and knowledge-based approaches for such types of data is quite challenging [3]. Also, finding abnormal patterns in large log-based data is extremely challenging due to the complex non-linear relationships among the components, processes, and sub-systems [4]. Due to the robust aircraft safety measures, unplanned breakdowns rarely occur during stable operation, but it is always high if it does occur the cost. Unplanned aircraft component failures' rare occurrence creates skewness or imbalanced distribution in the generated dataset [2,3]. Learning from the imbalanced data has been shown to degrade data-driven models' performance, causing unreliable prognostics [5,6]. There are many approaches to handling imbalance classification problems, but their application is highly tied to the application domain [7]. Therefore, in this study, we focus on tackling the imbalanced classification problem in the context of aircraft predictive maintenance modelling. Many studies have been conducted to modelling log-based data for predictive maintenance with varying levels of success [8]. Classical statistical predictive models have shown to be ineffective in handling imbalance classification in large log-based datasets because of the extreme rarity of some failures, complex hierarchical structure of the nomenclature of the failure type, the temporal features and the complex correlations of multivariate variables [8]. However, the recent advances in machine learning research, especially the capability of using deep neural networks to learn more complex temporal features, make it suitable for the large log-based dataset.

The remainder of this paper is structured as follows. Section 6.2.2 discusses the related work of this study. Section 6.2.3 provides a detailed architecture of the auto-encoder and BGRU. Section 6.2.4 shows the detailed architecture of the proposed deep learning hybrid method, which involves the integration of auto-encoder with BGRU. It further presents the experimental setup and the input data for the proposed model. The experimental result is presented and discussed in section 6.2.5. Finally, section 6.2.6 presents the conclusion and further work

6.2.2. Related Work

Rare failure detection and prediction is an active research field that has motivated the development of diverse methods. It is a fascinating and critical issue that has been approached within various contexts by research areas, such as machine learning and statistics [9]. Most of the approaches are specific to the application domain and the nature of datasets used as input. One of the approaches to identifying and predicting rare failures is to use a supervised machine learning approach, which can be framed in the form of anomaly detection, where the data is divided and labelled as negative and positive samples. The low dimensional features of the negatively labelled sample are extracted from higher dimension data using any of the feature extraction processes. Then rare failures are detected and predicted based on the reconstruction error. Most of the well known traditional or typical feature reduction and fault detection methods are the Principal Component Analysis (PCA), Partial Lease Square (PLS), and Independent Component Analysis (ICA). These methods use different ways to reduce data dimensionality, and they have achieved a varying degree of success on different data distributions. However, they have fundamental limitations to the non-linear features since they rely on linear techniques. Kernel tricks have been developed to help in converting the non-linear row data into linear data; examples are the KPCA [10] and KICA [11]. However, they require high computational power due to kernel function, especially if the data is large.

Although the approaches mentioned above have successfully handled normal fault detection and prediction, most of them still perform poorly when the target failures are extremely rare. Hence special techniques are needed to handle such cases. Some advances have been made to tackle rare event prediction, especially in aerospace domain predictive maintenance applications. For example, rare failure prediction using aircraft operational data can be seen in [12,13]. There are many existing approaches in the literature; some are suitable for solving failure detection and prediction in specific types of equipment. However, the particularities of our data limit us from using off the shelf approaches. Our approach differs from the aforementioned approaches in many aspects. We focus on predicting extremely rare failures using log-based aircraft central maintenance system data. The application of a hybrid deep learning technique that combines auto-encoder with bidirectional gated recurrent neural network model is proposed and developed for detecting and predicting extreme aircraft component replacement. The hybrid method will address the challenge of irregular patterns and trends caused by skew data distributaries, which will enhance the prediction of rare failures.

6.2.3. Autoencoder and Bidirectional Gated Recurrent Unit Network Architecture

This section presents auto-encoder and bidirectional gated recurrent unit network architecture and how these architectures are integrated to achieve better performance on large log-based, multivariate, non-linear, and time-series dataset.

6.2.3.1 Auto-encoder Architecture

Auto-encoder (AE) is a specific type of multilayer feedforward neural network that utilizes the back-propagation learning algorithm with respect to the loss function where the input is the same as the output neurons [14,15]. AE aims to learn the internal representation of the original data by compressing the input into a lower-dimensional space code, also called the latent-space representation, as shown in Figure 6-16. It then uses the compress representation to reconstruct the output while minimizing the error for the input data. Basically, AE comprises three components: Encoder X, latent-space P, and Decoder Y. The encoder compresses the input and produces the latent representation. The decoder then reconstructs the input only using the latent representation. An encoder with more than one hidden layer is called a deep auto-encoder.

The encoding and decoding process can be represented using the equation as follows.

$$p_i = f(w_p .\ x_i + \ b_t) \qquad (6\text{- }13)$$

$$y_i = g(w_y .\ p_i + \ b_t) \qquad (6\text{- }14)$$

Where f(.) and g(.) are the sigmoid functions, w_i represents the weights and b_i represents biases. The following minimized loss function is used to train the model.

$$L(X, Y) = \frac{1}{2\mathrm{n}} \sum_i^n \| \ x_i - \ y_i \|^2 \qquad (6\text{- }15)$$

Where x_i represent the observed value, y_i represent predicted values, and n represent the total number of predicted values.

Equation (6-13) helps in checking the validity of the resulting underlying feature P.

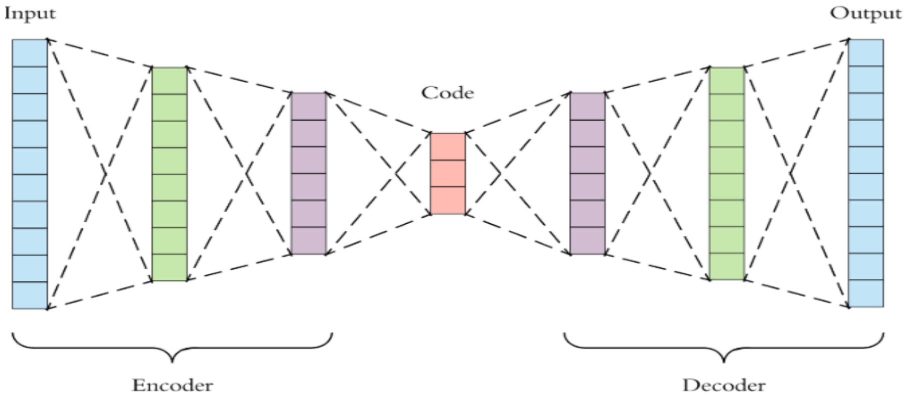


Figure 6- 15 Auto-Encoder Architecture[15]

Figure 6-15 shows a more detailed visualization of an auto-encoder architecture. First, the input data passes through the encoder, which is a fully-connected Artificial Neural Network (ANN), to produce the middle code layer. Then the decoder, which has a similar ANN structure, will produce the output only using the middle code layer. The goal is to get an output identical to the input.

Creating many encoder layers and decoder layers will enable the AE to represent a more complex input data distribution.

6.2.3.2 Bidirectional Gated Recurrent Unit Network Architecture

A Bidirectional Gated Recurrent Unit (BGRU) is a recurrent neural network that is successfully used to solve time series sequential data problems [16]. Each BGRU block contains a cell that stores information. Each block is made up of a reset and update gate, and the cells help tackle the vanishing gradient problem. The reset gate determines how to combine new input with previous memory, while the update gate defines how much of the previous memory to keep around. BGRU comprises two GRU blocks. The input data is fed into the two networks, the feedforward and feedback with respect to time, and both of them are connected to one output layer [17]. The gates in bidirectional GRU are designed to store information longer in both directions, providing better performance than feedforward networks. This provided a capability to use both the past and future contexts in a sequence. BGRU can be expressed as

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \qquad (6\text{- }16)$$

$$where\ \overrightarrow{h_t},\ is\ the\ feedfoward\ \ and\ \overleftarrow{h_t}\ the\ backward\ block$$

The final output layer at time t is

$$y_t = \sigma(W_y h_t + b_y) \qquad (6\text{- }17)$$

Where σ is the activation function W_v is the weight, $and\ b_v$ is the bias vector.

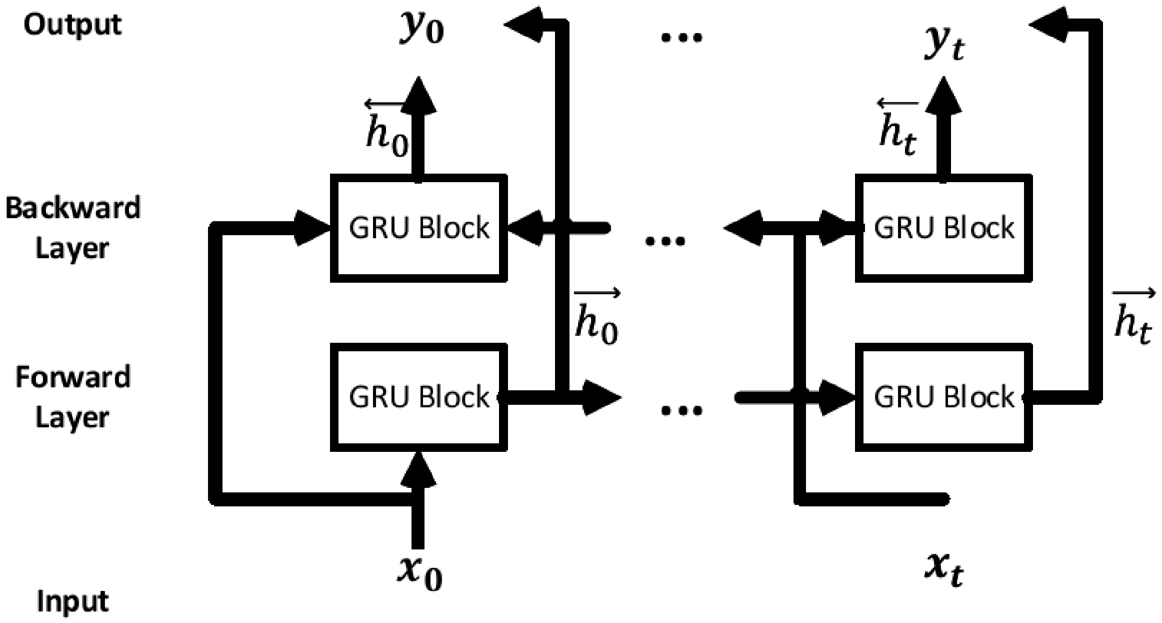


Figure 6- 16 BGRU architecture with forward and backward GRU layers

As seen in Figure 6-17 expanded in Figure 6-18, each of the GRU blocks is made up of four components. Input vector x_I with corresponding weights and bias, reset gate r_I with corresponding weight and bias $W_{r,}U_r, b_r$, update gate z_I with corresponding weight and bias $W_{z,}U_z, b_z$, and out vector h_t with its weight and bias $W_{h,}U_h, b_h$. Fully gated unit is represented as follows

Initially, for t = 0, the output vector is h_0 = 0

$$z_t = \sigma_g\ (W_z x_t + U_z h_{t-1} + b_z) \qquad (6\text{- }18)$$

$$r_t = \sigma_g\ (W_r x_t + U_r h_{t-1} + b_r) \qquad (6\text{- }19)$$

$$h_t = z_t\ h_{t-1} + (1 - z_t) \otimes \emptyset\ h\ (W_h x_t + U_h\ (r_t \otimes h_{t-1}\ ) + b_h) \qquad (6\text{- }20)$$

Were ⊗ is the Hadamard product. W, U, b are parameter matrices and vector. $\sigma_g\ and\ \emptyset h$ are the activation functions, $\sigma_g\ is\ a\ sigmoaid\ fuction\ and\ \emptyset h\ a\ hyperbolic\ tangent.$

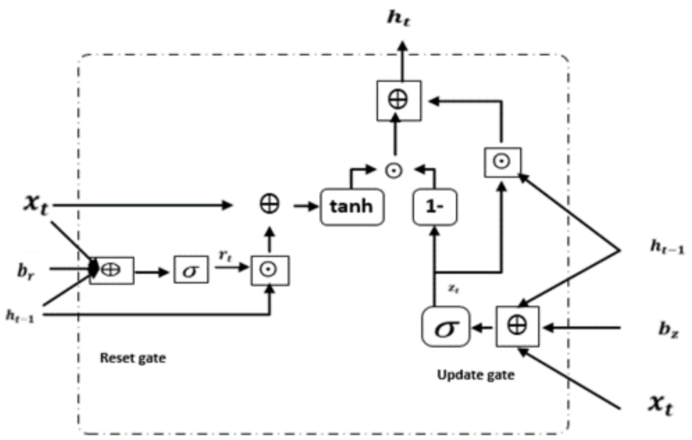


Figure 6- 17 Shows a GRU block (See Figure 6-17) with an update and reset gate, sigmoid and hyperbolic tangent

6.2.4. Proposed Model

6.2.4.1. AE-BGRU Network

Our objective is to develop a model that detects and predicts extremely rare failures from the large log-based dataset for practical aircraft predictive maintenance. The basic idea is to separate the rare failure detection problem from the prediction problem. Therefore, the proposed model employs two stages, detecting rare failure using auto-encoder and predicting the next occurrences of the detected failure in the next-N- step using BGRU.

The dataset is extremely imbalanced. The imbalanced ratio between the positively labelled and negatively labelled data is less than 5%. Deep learning techniques are limited because of the overwhelming majority class in such an extremely rare failure problem. This scenario affects the predictive models' accuracy in general and the normal dropout, and batch normalization is ineffective in extremely imbalanced problem scenarios [3,18]. Also, the use of under-sampling and over-sampling methods to balance the dataset is not suitable for the type of dataset considered in this study because, apart from its limitation of low accuracy, it is also not practical to alter the original structure of the data, as the target is to learn the exact patterns that lead to such failures. Therefore, we proposed AE-BGRU to handle the problem differently. The structure of the proposed model is shown in Figure 6-19.

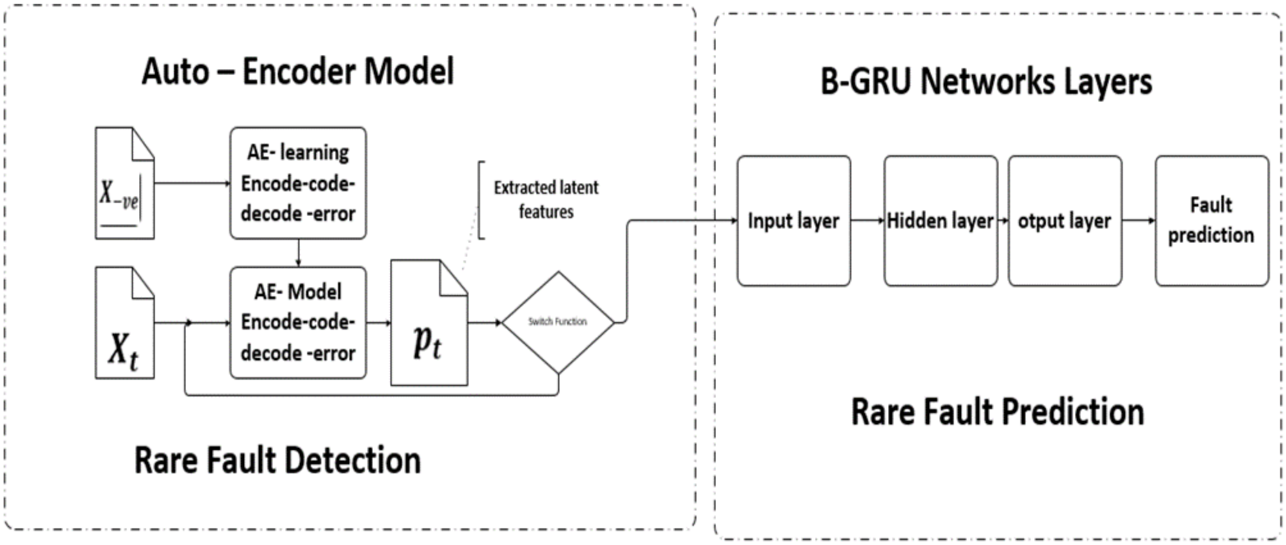


Figure 6- 18 A structure of the integrated auto-encoder and bidirectional gated recurrent unit networks for rare fault detection and prediction

We, first of all, use AE to detect the rare failure using reconstruction errors. We divide the data into positive labelled (rare class) and negatively labelled (majority class). As seen in Figure 6-19, the first section of AE trained with only negatively labelled data X_{-ve} in the process of training, the highest reconstruction error is determined and then used as the threshold for the AE-BGRU Model. The AE is trained by feeding the encoder layer with the original negatively labelled data, and its latent feature is extracted in the middle latent code layer. Then the decoder layers try to reconstruct the original using compressed latent features as input. In the encode-decode process, a reconstruction error is determined. Since the AE model is learned using negatively labelled data, any new data point at the AE-BGRU model can easily be detected. Any example from the same negatively labelled class is expected to have low error since it has the same distribution. However, when a new data point from a positively labelled class is encountered, it is expected to have a higher reconstruction error score. For example, an observation say $x_t \epsilon X_t$ will be classified as a fault if the reconstruction error exceeds the normal else, it will be classified as no-fault. Once the faults are discovered, the resulting data is seamlessly fed into the BGRU section of the AE-BGRU model for the failure prediction. The input data to the BGRU model is sequential; it includes the data before detecting the first fault, composite with the time delay, and the false positive of the AE.

The BGRU section of the model is designed as follows. First, the BGRU cells are constructed so that the result of feedforward is computed (F_t) and the feedback propagation (B_t) are merged at the first BGRU layer. Four different methods can be used to merge the outcome, concatenation (default), summation, multiplication, and average. The merging is represented as follows

$$O_t^1 = concat\,((\overrightarrow{F_t}), (\overleftarrow{B_t})) \tag{6- 21}$$

Such that$(\overrightarrow{F_t}) = (\overrightarrow{h_1}, \overrightarrow{h_2}, \overrightarrow{h_3}, \dots, \overrightarrow{h_t})$

And $(\overleftarrow{B_t}) = (\overleftarrow{h_t}, \overleftarrow{h_{t+1}}, \overleftarrow{h_{t+2}}, \overleftarrow{h_{t+3}}, \dots\overleftarrow{h_n},)$

Second, a fully connected layer is used to multiply the BGRU network's output with its weight and bias. Then a softmax regression layer makes a prediction using input from the fully connected layer. A weighted classification layer is used in the BGRU to compute the weighted cross-entropy loss function [19] for prediction score and training target, which tackles the imbalanced classification problem. The following loss is used,

$$\mathrm{H}(p_{,t}) = -\sum(1 - (p_x))^{\gamma} log_2\,(t_x) * \theta_i \tag{6- 22}$$

Where H($p_{,t}$) represent the estimated probability of each class, p is the target distribution, and t is approximating the target distribution. $\gamma \geq 0$ is the discount factor parameter that can be tuned for best estimation and θ_i is the logic weight of each class.

6.2.4.1 Case Study and Experimental Setup

In this paper, we have evaluated the detection and prediction of individual aircraft component replacement. The experiment is set up to verify the performance of our proposed approach in handling the rare occurrences of failure compared to the state of the art, deep learning approaches for rare event predictions. The fundamental research question of interest is then whether AE-BGRU, with explicit failure detection and additional training capability, can outperform the normal unidirectional deep learning time-series methods on an extremely imbalanced dataset. Another important question is can model performance for rare failure prediction be improved if learning is done in two directions (feedforward and feedback propagation). Also, how different do the architecture of deep learning models treat the input data? To investigate the above questions, we conduct a series of experiments and report the result. We categorize the modelling approach into two; binary class and multi-class. In the first scenario, we modelled it as a multi-class classification problem that predicts all the targeted component failures at the same time. Secondly, we modelled it as a binary classification problem that predicts individual functional items. In both cases, the prediction is to consider 10 flight legs in advance and not later than two legs to failure.

6.2.4.2 Dataset

This study uses over eight years' worth of data. The datasets are collected from two databases. The first database is the Aircraft Central Maintenance System (CMS) data, which comprises error messages from BIT (built-in test) equipment (that is, aircraft fault report records) and the flight deck effects (FDE). These messages are generated at different stages of flight phases (take-off, cruise, and landing). The second database is the logs of aircraft maintenance activities (i.e. the comprehensive description of all recorded aircraft maintenance activities). The dataset used in this study is obtained from a fleet of long-range (A330) aircraft. We choose a target functional item Number (FIN) of high practical value with an adequate number of known failure cases. We consider three specific components that are replaced due to unscheduled maintenance and study their failure behaviours. The behavioural patterns are then used to build a predictive model for predicting replacement. Data from the year 2011 to 2016 is used for training, while the reaming from 2016 to 2018 is used for testing. The targeted Line-replacement Unit (LRU's) from A330 –long-range (LR) aircraft family are 4000KS - Electronic Control Unit/ Electronic Engine Unit, 4000HA – Pressure Bleed Valve, and 438HC – Trim Air Valve.

6.2.5 Result and Discussion

Binary Classification approach- predicting individual FI's

Result based on performance comparison with basic deep learning approaches for time series and rare event predictions.

Table 6- 3 Aircraft A330 rare failure prediction of individual LRU's using CMS dataset

	Precision	Recall	G-mean
LSTM	4000KS = 0.69 4000HA = 0.72 438HC =0.77	4000KS =0.72 4000HA = 0.70 438HC =0.69	4000KS =0.68 4000HA = 0.70 438HC = 0.63
GRU	4000KS =0.63 4000HA = 0.69 438HC = 0.67	4000KS = 0.82 4000HA = 0.65 438HC = 0.61	4000KS =0.65 4000HA = 0.62 438HC = 0.62
AE-BGRU	4000KS = 0.92 4000HA = 0.87 438HC = 0.88	4000KS =0.89 4000HA = 0.81 438HC = 0.80	4000KS =0.66 4000HA =0.63 438HC = 0.65

We have investigated if training of an imbalanced dataset in two-direction can improve model performance. As seen in Table 6-2, the proposed model shows the superior performance when comparing the proposed AE-BGRU model and the normal GRU and LSTM in predicting rare replacement of aircraft LRU replacements. For the three FIN, the AE-BGRU model shows approximately an overall improvement of 25% in precision, 14% in a recall, and 2% in G-mean.

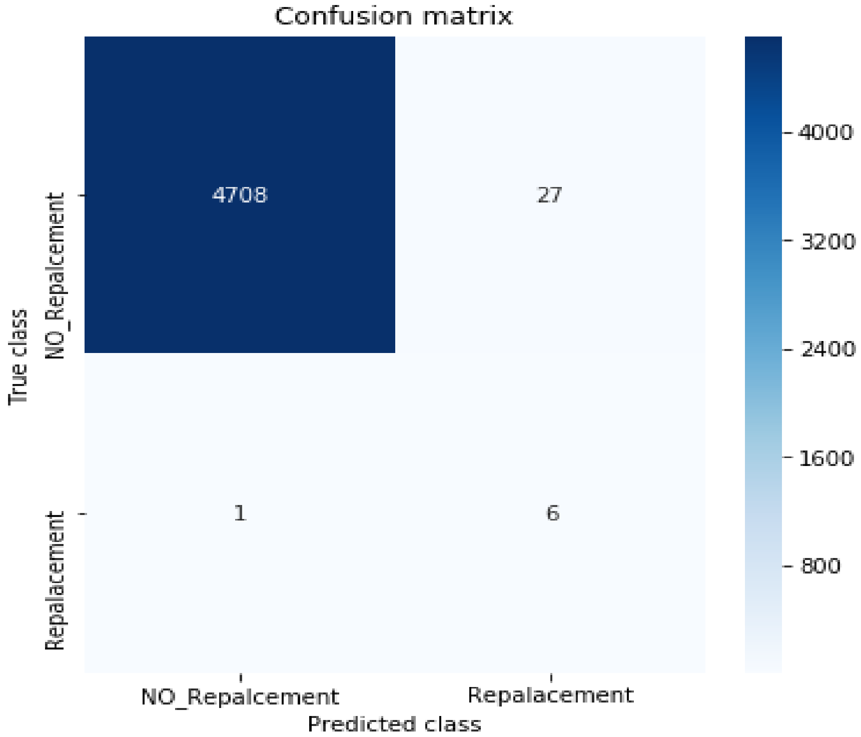


Figure 6- 19 Confusion matrix on test prediction for FIN_4000KS using a reconstruction error threshold of 0.4

Figure 6-20 shows that AE-BGRU predicted 6 out of 7 unplanned electronic engine unit (FIN_4000KS) replacements. This prediction includes 10 flight legs in advance, and it can also be observed that the model detected and predicts approximately 85% of extremely rare failures, which is a reasonable specificity, especially for aircraft maintenance.

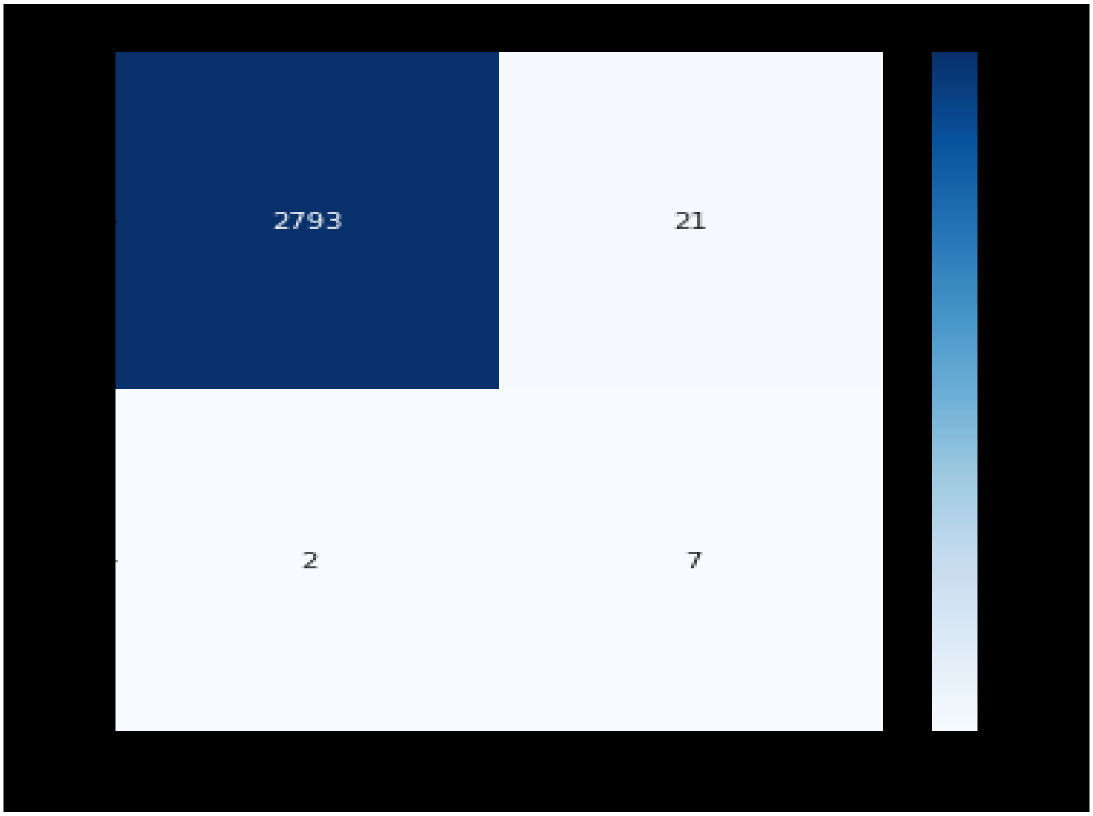


Figure 6- 20 Confusion matrix on test prediction for FIN_4000HA using a reconstruction error threshold of 0.4

Figure 6-21 shows that AE-BGRU predicted 7 out of 9 unplanned replacement of pressure bleed valve (FIN_4000HA) failures. This prediction includes 10 flight legs in advance, and it can also be observed that the model detected and predicts approximately 80% of extremely rare failures, which is a reasonable specificity, especially for aircraft maintenance.

6.2.5.1 General Discussion:

The AE-BGRU model shows an overall improvement of 25% precision, 14% recall, and 2% G-mean for the three FIN considered. The superior performance indicates that AE-BGRU model networks are able to capture the underlying temporal structure better by traversing the input data twice (feedforward and feedback) in making the prediction. Furthermore, performance improvement of AE-BGRU against the normal is understandable for certain types of data, such as in-text classification and prediction of text-words in a sequence to sequence learning. However, it was not clear whether training extreme imbalanced, numerical, time-series data using a bidirectional approach would improve model performance as there might not exist some definite temporal contexts and observable in-text sequence learning. Therefore, our results show that AE-BGRU outperformed normal GRU and LSTM even in the context of predicting rare failure in log-based aircraft CMS datasets.

6.2.6. Conclusion and Future Work

This paper proposes a novel technique, AE-BGRU, that can narrow down the volume of logs of aircraft warning or failure messages into a small set of important and most relevant logs. AE-BGRU uses the integration of auto-encoder with bidirectional gated recurrent networks, which complement each other to generate accurate link failure/warning messages in relation to aircraft LRU removal. The auto-encoder helps in training the model with only negatively labelled data to detect rare faults using the reconstruction error as a threshold. The output of AE is used as input to the BGRU network to predict the occurrence of those faults in the Next-N-step. We have implemented this technique and applied it to the problem of rare failure detection and prediction. Our evaluation indicates that AE-BGRU can effectively find the important log messages that hold direct links to aircraft LRU failure causes, leading to replacement. We demonstrate the concepts, design, and evaluation results using real-world aircraft central maintenance system log-based data. Comparing AE-BGRU with other similar deep learning methods, the proposed approach is 25% better in precision, 14% in recall, and 3% in G-mean. The result also shows robustness in predicting failure within a defined useful period. Further studies will be conducted on other architectures of AE-CNN-BGRU to improve prediction performance on the aircraft log-based dataset to achieve predictive maintenance.

6.2.7 Acknowledgement

Appreciation goes to Dr S.P.King CEng, FIET, FIMA [Visiting Professor – Cranfield University], EHM Specialist & Rolls-Royce Engineering Associate Fellow – Advanced Analytical Methods, Civil Digital Solutions – C&S, Rolls-Royce plc for providing support. To the Integrated Vehicle Health Management Centre (IVHM), Cranfield University, I am given the opportunity to carry out this study and the PTDF- Nigeria for sponsoring the study.

6.2.8 References

1. Dai X., Gao Z. From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis. IEEE Transactions on Industrial Informatics. IEEE; 2013; 9(4): 2226–2238. Available at: DOI:10.1109/TII.2013.2243743

2. Saufi SR., Ahmad ZA Bin., Leong MS., Lim MH. Challenges and Opportunities of Deep Learning Models for Machinery Fault Detection and Diagnosis: A Review. IEEE Access. IEEE; 2019; 7: 122644–122662. Available at: DOI:10.1109/access.2019.2938227

3. Park P., Di Marco P., Shin H., Bang J. Fault detection and diagnosis using combined autoencoder and long short-term memory network. Sensors (Switzerland). 2019; 19(21): 1–17. Available at: DOI:10.3390/s19214612

4. Dangut M david., Skaf Z., Jennions IK. An integrated machine learning model for aircraft components rare failure prognostics with log-based dataset. ISA Transactions. Elsevier Ltd; 2020; (xxxx). Available at: DOI:10.1016/j.isatra.2020.05.001

5. Raghuwanshi BS., Shukla S. UnderBagging based reduced Kernelized weighted extreme learning machine for class imbalance learning. Engineering Applications of Artificial Intelligence. Elsevier Ltd; 2018; 74(July): 252–270. Available at: DOI:10.1016/j.engappai.2018.07.002

6. Wu Z., Lin W., Ji Y. An Integrated Ensemble Learning Model for Imbalanced Fault Diagnostics and Prognostics. IEEE Access. 2018; 6: 8394–8402. Available at: DOI:10.1109/ACCESS.2018.2807121

7. Ali A., Shamsuddin SM., Ralescu AL. Classification with class imbalance problem: A review. International Journal of Advances in Soft Computing and its Applications. 2015; 7(3): 176–204.

8. Burnaev E. Rare Failure Prediction via Event Matching for Aerospace Applications. 2019; (July). Available at: http://arxiv.org/abs/1905.11586

9. Berberidis C., Angelis L., Vlahavas I. Inter-transaction association rules mining for rare events prediction. Proc. 3rd Hellenic Conference …. 2004; Available at:

http://lpis.csd.auth.gr/publications/076-Berberidis-Angelis-Vlahavas-SETN04.pdf

10. Kallas M., Mourot G., Anani K., Ragot J., Maquin D. Fault detection and estimation using kernel principal component analysis. IFAC-PapersOnLine. 2017; 50(1): 1025–1030. Available at: DOI:10.1016/j.ifacol.2017.08.212

11. Lee J-M., Qin SJ., Lee I-B. Fault Detection of Non-Linear Processes Using Kernel Independent Component Analysis. The Canadian Journal of Chemical Engineering. 2008; 85(4): 526–536. Available at: DOI:10.1002/cjce.5450850414

12. Alestra S., Bordry C., Brand C., Burnaev E., Erofeev P., Papanov A., et al. Rare event anticipation and degradation trending for aircraft predictive maintenance. 11th World Congress on Computational Mechanics, WCCM 2014, 5th European Conference on Computational Mechanics, ECCM 2014 and 6th European Conference on Computational Fluid Dynamics, ECFD 2014. 2014. pp. 6571–6582.

13. Burnaev E. Rare Failure Prediction via Event Matching for Aerospace Applications. 2019 3rd International Conference on Circuits, System and Simulation, ICCSS 2019. IEEE; 2019; : 214–220. Available at: DOI:10.1109/CIRSYSSIM.2019.8935598

14. Baldi P. Autoencoders, Unsupervised Learning, and Deep Architectures. ICML Unsupervised and Transfer Learning. 2012; : 37–50. Available at: DOI:10.1561/2200000006

15. Le Q V. A Tutorial on Deep Learning Part 2: Autoencoders, Convolutional Neural Networks and Recurrent Neural Networks. Tutorial. 2015; : 1–20. Available at: DOI:https://cs.stanford.edu/~quocle/tutorial2.pdf

16. Farzad A., Gulliver TA. Log Message Anomaly Detection and Classification Using Auto-B/LSTM and Auto-GRU. 2019; : 1–28. Available at: http://arxiv.org/abs/1911.08744

17. Savoy J., Gaussier E. Information retrieval. Handbook of Natural Language Processing, Second Edition. 2010. 455–484 p. Available at: DOI:10.4324/9781351044677-24

18. Bengio Y., Courville A., Vincent P. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence. IEEE; 2013; 35(8): 1798–1828. Available at: DOI:10.1109/TPAMI.2013.50

19. Murphy KP. Machine Learning A Probabilistic Perspective. The MIT Press. 2012. Available at: DOI:10.1007/978-94-011-3532-0_2

6.3 Rescaled-LSTM for Predicting Aircraft Component Replacement Under Imbalanced Dataset Constraint

Deep learning approaches are continuously achieving state-of-the-art performance in aerospace predictive maintenance modelling. However, the data imbalance distribution issue is still a challenge. It causes performance degradation in predictive models, resulting in unreliable prognostics, which prevents predictive models from being widely deployed in real-time aircraft systems. The imbalanced classification problem arises when the distribution of the classes present in the datasets is not uniform, such that the total number of instances in a class is significantly lower than those belonging to the other classes. It becomes more challenging when the imbalance ratio is extreme. This paper proposes a deep learning approach using re-scaled Long Short Term Memory (LSTM) modelling for predicting aircraft component replacement under imbalanced dataset constraints. The new approach modifies each class's prediction using a re-scale weighted cross-entropy loss, which controls the majority classes' weight to have less contribution to the total loss. The method effectively discounts the effect of misclassification in the imbalanced dataset. It also trains the neural networks faster, reduces over-fitting and makes a better prediction. The results show that the proposed approach is feasible and efficient, achieving high performance and robustness via skewed aircraft central maintenance datasets.

6.3.1 Introduction

The technological growth in the aerospace industry and the continued advancement in data analytics have made the generation and analysis of large quantities of aircraft data more affordable. This has caused a transformation in maintenance strategies, such as shifting from preventive maintenance to predictive maintenance. Research into developing data-driven prognostic models for condition-based maintenance is gaining more attention [1,2]. However, one of the major problems researchers face is the low representation of faulty asset behaviour, which results in an imbalanced classification problem [3]. This problem arises when the distribution of classes present in the dataset is not uniform, such that the total number of instances in one class far outnumber that of the other class. This degrades the performance of the data-driven model, causing imprecise prognostics. Therefore, solving this problem is still an open issue[4–6]. The imbalanced classification problem is prevalent in many application domains. For example, in aircraft' predictive maintenance, the historical data is often imbalanced because the aircraft component replaced due to unscheduled maintenance is most time rare in the overall maintenance records database [7]. The data-imbalanced problem can also be seen in financial fraud. The illegitimate transactions are rare compared to legitimate ones. It is critical to detect the rare or minority class examples because failure to detect any fraud the consequence can be grave [8]. Similarly, imbalance learning has application in clinical science for the diagnostic of rare diseases. In most cases, the infected population is rare compared to a healthy population [9]. Likewise, in detecting oil spillage in the ocean, images obtained by satellite may show a few images representing the oil spillage portion, while most of the images representing the non-spillage areas and the interest are to identify the minority [10], and much more application domain.

Several research approaches have been conducted to solve the imbalanced classification problem. Some comprehensive reviews about the imbalance problem can be found in [11–13]. The Imbalanced classification problem's solution can be categorised into three main approaches: the data level, the algorithm level, and the hybrid approach (see Figure 6-22. The data level approach involves re-sampling the dataset before presenting it as an input to the learning algorithm. The algorithm level approach tackles the imbalanced learning problem by changing the learning algorithm to respond favourably to both classes during learning[12]. In contrast, the hybrid combines two or more to achieve better performance.

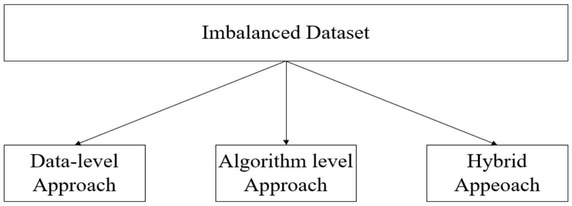


Figure 6- 21 The three ways of the handling imbalance problem

6.3.1.2 Deep learning – LSTM Architecture

Long Short Term Memory (LSTM) network is a special kind of recurrent neural network capable of learning long-term and short-term dependencies. They are used to model time-series or sequence-dependent variables, such as machine failure records, electricity consumption, stock market price, and so on. LSTM was first introduced by [14], specifically designed to overcome the recurrent neural network's long-term dependency problem. The networks work pretty well on a variety of problems and are widely used recently.

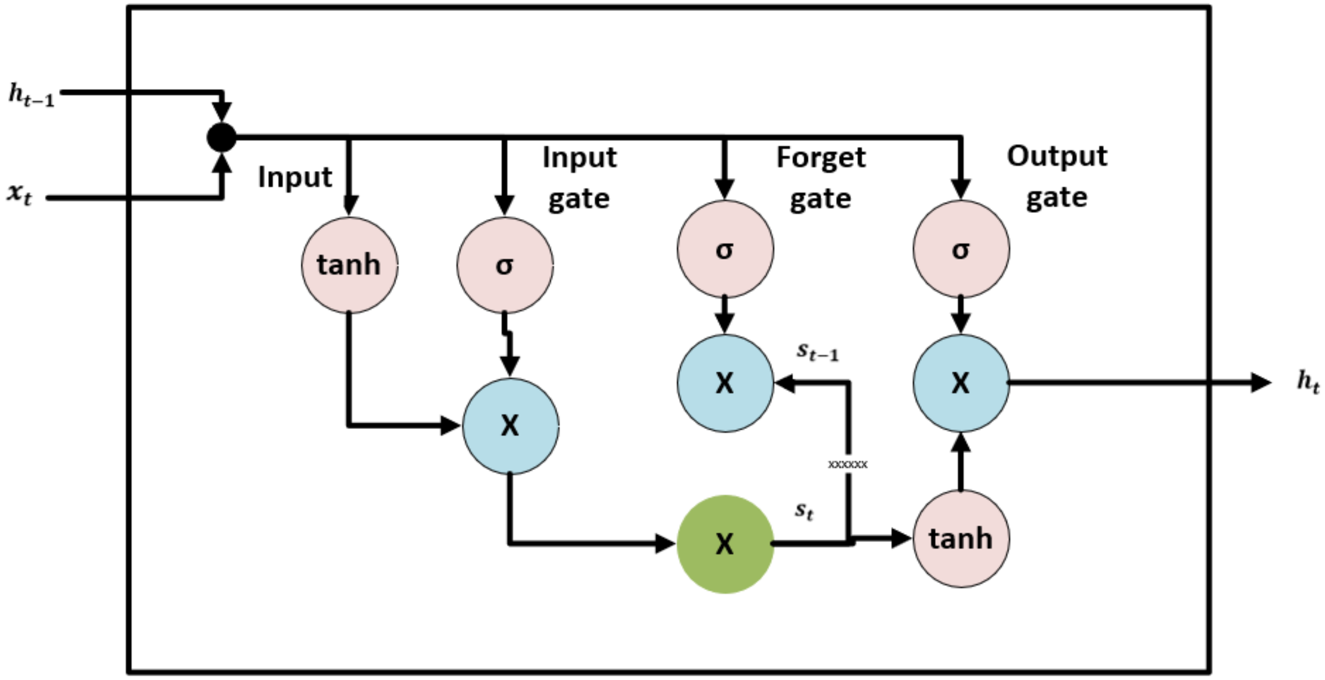


Figure 6- 22 Structure of LSTM Network

From Figure 6-23, on the left-hand side, we have a new sequence value x_t which combine with the output from the previous cell h_{t-1}. The initial step for the combination input in the new cell is for it to be dense using $tanh$ layer. The second derivative of $Tanh$ activation function can be sustained for a long-range before descending to zero. Therefore, it is suitable for handling the challenge of vanishing gradient. Secondly, the input data is passed through the input gate layer using a sigmoid activation function, whose output is joined by dense layer input. The input gate filters out any unwanted element of the input vector. The sigmoid function is a gate function for the three gates – the input, output and forget. It outputs values between 0 and 1. It either allowed flow or disallowed the flow of information throughout the gates. Finally, the output gate determines which values are actually allowed as output from the cell ht.

6.3.1.3 The main contribution

Therefore, this paper's contribution is in the development of a data-driven deep learning predictive model to predict aircraft component replacement under imbalanced data constraints. Second, we conducted a comparative experiment to find a loss function suitable for handling an imbalanced dataset. A new approach is proposed to integrate the re-scaled weight loss function into the LSTM model to optimise classification in the imbalanced dataset. The model is evaluated on real-life Aircraft Central Maintenance (CMS) datasets and proves its robustness.

The main advantage of our proposed deep learning approach over other state-of-the-art neural network techniques for imbalanced classification are:

a. Our approach supports the efficient learning of temporal dependencies, and it can handle non-linearity and volatility dynamics features in time-series datasets, which improve the performance of prediction.

b. The new approach presents a unique way of changing loss function with respect to weights and a unique arrangement of LSTM networks. The new strategy dynamically regulates the combined weight to produce a merged predicting result. The LSTM model weights are combined at each time step adaptively and recursively by using both the errors of past predictions and discarded weight at the forget gate layer.

c. The proposed approach is computationally efficient because it uses a simple optimisation method while finding and combining model weights.

6.3.2 Related Work

This section provides research work that has used the LSTM neural network to address imbalanced classification, focusing on aircraft' predictive maintenance. Many research efforts have been made in deep learning to address the imbalanced classification problem. The majority of the work still falls under the three categories of handling an imbalanced dataset (data level, algorithm level, and hybrid approach). The difference is in the implementation of those approaches on a neural network. For example, [15] show the effects of an imbalanced classification problem as a course of slow convergence in the neural network backpropagation algorithm. The authors present a modified technique for calculating a direction in weight-space, decreasing the error for each class, thus addressing the imbalanced problem. Furthermore, [16] show how cost-sensitive can be implemented in neural networks by modifying the backpropagation learning algorithm for multi-layered feed-forward neural networks.

[17] shows the effect of data sampling and threshold-moving in training cost-sensitive neural networks. The authors use manoeuvred threshold toward the minority classes such that examples with higher costs become harder to misclassify, hence getting better prediction. [18] this study provides a general review of existing deep learning techniques for addressing class imbalanced data. Other works that have shown how imbalanced classification can be handled in training neural networks are [19–23]. [24] shows the use of LSTM to predict multiple site fatigue damage prediction of aircraft lap joints. [25] proposes a weighted deep representation learning model for imbalanced fault diagnosis in Cyber-Physical Systems, which uses under-sampling to balance the dataset and then design a weighted loss to optimised prediction. However, the use of random under-sampling to balance the dataset can be prone to losing informative data points as it is with any random under-sampling approaches. [26] presented a method of handling imbalanced data using neural networks. The method is designed to learn the embedding using a novel objective function, called triple-header cross-entropy, and they test it to detect acoustic event problems.

Despite these advances to solve the data imbalance problem in a neural network, the open literature lacks an exhaustive unified solution to generally handle predictive modelling. In fact, many researchers agree that the subject of deep learning with imbalanced data is understudied. Hence, it is still an open area of research.

6.3.3 Problem Formulation

We focus on the deep neural network approach to address the data imbalance problem during the training.

To deal with the aforementioned challenge, we use an algorithm level approach to handling the imbalanced classification problem. We modify the LSTM network by re-scaling the loss function to be robust and efficient in handling hard to learn examples (examples from minority class). This also enhances the computational efficiency of LSTM, reduce overfitting, reduce false positive and false negative rate.

Deep learning LSTM has been widely used in the analysis of time series datasets, image classification, natural language processing, time series, and many more [27–30]. However, apart from the challenge of imbalanced classification, many challenges arise when the network becomes deep. Such that their convergence becomes slow. The deep LSTM network is prone to the vanishing gradient and gradient exploding (where gradients slowly disappear as we back-propagate across multiple network layers). This makes training very difficult, although this can be mitigated if gradient clipping is properly set. The other challenge is that it cannot easily be stacked into the depth network layers because the saturated activation function used makes the gradient decay over layers; thus, accuracy can easily fall.

6.3.3.1.Proposed method: Re-scaling cross-entropy loss

In machine learning or optimisation in general, a loss function can be defined using cross-entropy (CE). CE is mostly used to quantify a variance between two probability distributions. Considering a binary classification that involves classifying data into two possible classes, say 0 and 1. The model can predict an output of the form $y_x \in \{0,1\}$ given the input vector x. For example, given training instance with label y_3 out of possible labels y_1, y_2, and y_3. The ideal distribution for this case will be Prb(y_1) = 0.0, Prb(y_2) = 0.0 and Prb(y_3) = 1.0, which can be interpreted as the given training instance has the 0% probability of being a class y_1 and class y_2 and 100% probability of being a class y_3. But if a machine-learning algorithm predicts probability distribution as follows: Prb(y_1) = 0.05, Prb((y_2) = 0.15 and Prb(y_3) = 0.8, in this case, the CE will determine how close is the predicted distribution to the true distribution using $H(p,t) = -\sum p(x) log t(x)$ Where p is the wanted probability and t the actual probability.

Logistic function $g(h) = \frac{1}{1+e^{-2\beta h}}$ (6- 23)

Is used to model the probability of each given input. Where a probability of getting an output

$y = 1$ is given as $t_{y=1} = \dot{y} \equiv g(w.x) = {}^{1}/_{(1+e^{-w.x})}$ (6- 24)

where the vector of weight w is optimised through stochastic gradient descent while the probability of getting an output

$y = 0$ is $t_{y=0} = 1 - \dot{y}$ (6- 25)

The true probabilities can be written as

$$(p, x) = \begin{cases} p_{y=1} = y \\ p_{y=0} = 1 - y \end{cases} \tag{6- 26}$$

let $p \in \{y, 1 - y\}$ and $t \in \{\hat{y}, 1 - \hat{y}\}$ we can use cross-entropy to measure dissimilarity between p and q such that

$$H(p, t) = -\sum_i p_i log t_i = - y\ log\hat{y} - (1 - y) \log{(1 - \hat{y}\}}\ ) \tag{6- 27}$$

to compute the loss function, we take the cumulative average of all the cross entropies. For instance, if N is the total number of data points with each sample index by $= 1, \ldots, N$. then the loss function is given by

$$\boldsymbol{K(w) = \frac{1}{N} \sum_{n=1}^{N} H(p_n, t_n)} \tag{6- 28}$$

$$= -\frac{1}{N} \sum_{n=1}^{N} [y_n log\hat{y}_n + (1 - y_n) \log{(1 - \hat{y}_n)}] \tag{6- 29}$$

Where $\hat{y}_n \equiv g(w.x_n) = {}^{1}/_{1\, +\, e^{-w.x_n}}$ with $g(z)$ as the logic function.

The cross-entropy method is a Monte Carlo method for optimisation and sampling importance in class distributions [31]. The minimisation cross-entropy methods have been used to optimise the rare event's prediction, as shown in [32]. Therefore, we can also optimise the cross-entropy method for imbalanced datasets by building on this approach. The cross-entropy method approximates the optimal importance sampling estimator iteratively as follows:

1. Draw a sample from a probability distribution.

2. Minimise the cross-entropy between this distribution and a target distribution to produce a better sample in the next iteration.

Therefore, to minimise the cross-entropy between distribution, we use estimation via class importance sampling and Kullback–Leibler (KL) divergence [33], which measure how one probability distribution differs from a second.

Therefore, considering the general problem of estimating the quantity.

Focal loss $\tau = \text{ב}_{\mu}[H(x)] = \int H(x) f(x, \mu) dx$ (6- 30)

Where H is the performance function, and $f(x, \mu)$ is a member of some parametric family of distribution. To estimate using importance sampling from (Eq 6-29) to get.

$\tau = \frac{1}{n} \sum_{i=1}^{n} H(x) \frac{f(x_i, \mu)}{g(x_i)}$ (6- 31)

Where $x_i, \dots, x_n$ is a random sampling from g for H^{+} then the optimal density is given by

$g^{*} = {}^{H(x) f(x, \mu)}\!/_{\tau},$ (6- 32)

However, this depends on the known τ. The goal of cross-entropy is to approximate the optimal probability distribution by adaptively choosing the number of the parametric that are closes to the optimal probability distribution g^{*} Using KL.

Cross-entropy loss is widely used because of its classification strength, that even easily classifiable examples (majority class data points) result in a significant loss [15]. During model training, the overall total error cost representing the majority samples negatively impacts the minority class samples because most of the majority class's losses will dominate the gradient, producing an undesired result. Therefore, to handle such challenges, [34] proposed modifying the normal CE loss function that down-weight samples from the majority samples. They contribute less to the total loss and focus more on the hard ones (the minority class). The archived that by introducing a term (1 − $(p_t)^{\gamma}$ in to the normal CE which controls biasness in the overall cost and enhance learning from hard to learn examples. A general way of formulating Focal Loss -FL is:

$(p_{,t})$ = −(1 − $(p_t)^{\gamma}$log(p_t) (6- 33)

Where $(p_{,t})$ represent the estimated probability of each class, and $\gamma \geq 0$ is the discount factor parameter that can be tuned for the best estimation. Therefore, to further improve the prediction of the minority class in the extreme imbalanced datasets, we modify the focal loss function further by re-scaling and multiplying logics to weight to have:

RFL$(p_{,t})$ = − (1 − (p_t))$^{\gamma}$ log_2 (p_t) * θ_i (6- 34)

Where θ_i is the logic weight of each class.

Using (eq 6-33) during training, weights are transformed from class weight to weight per example, and this increases the strength of predicting minority class samples.

6.3.4 Methodology

This section integrates a process of handling imbalanced fault prognostics and diagnostics into the traditional process of developing predictive models. The methodology comprises three stages data pre-processing, model training and model evaluation, as seen in Figure 6-24. The raw data is transformed into the right format for machine learning in the pre-processing data stage. We replace null values with zero; missing values in the data are ignored. Remove unwanted columns. The next step was transforming the time series data into a machine learning format to use machine learning for time-series prediction. The problem needs to be reframed as an unsupervised or supervised learning problem. This study frames the problem as supervised learning: having a pair of input-output structures. We divide the data into standard segments using a sliding window.

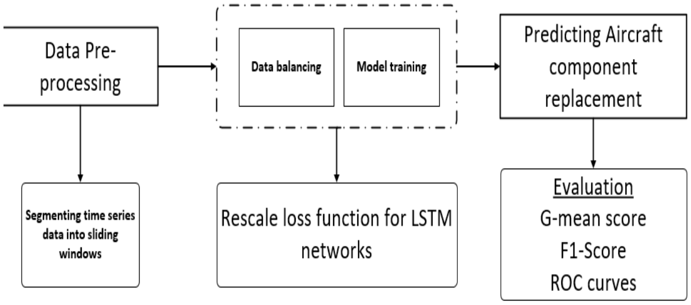

Data Pre-processing
Data balancing
Model training
Predicting Aircraft component replacement
Segmenting time series data into sliding windows
Rescale loss function for LSTM networks
Evaluation
G-mean score
F1-Score
ROC curves

Figure 6- 23 Flow of the R-LSTM Implementation

In the modelling stage, feature engineering is taken care of since we are using a deep learning approach. A novel modification of loss function in LSTM is presented, which incorporates multiple predictions from a set of individual neural networks. The network layers are integrated to determine the output data, reducing variance in the Imbalanced dataset and optimising prediction.

6.3.5 Experiments and Evaluations

In this study, two experiments are conducted as follows:

1. To compare the performance of existing loss functions with the proposed loss function.

This experiment is set up to compare the performance of our approach with the existing loss functions. We compare binary classification and multiclass loss functions with the proposed loss function for extreme imbalanced classification. In this experiment, a dataset is created using a machine learning sci-kit learn function make circles with a random number of samples, noise and varying imbalanced ratio.

2. Compare our approach with existing methods of handling imbalanced classification problems and basic deep learning approaches for time-series predictions.

This experiment is set up to verify our proposed approach in handling imbalanced datasets compared to the state of the art approaches.

Dataset: This study uses over eight years' worth of data. The datasets are collected from two databases. The first database is the Aircraft Central Maintenance System (CMS) data, which comprises error messages from BIT (built-in test) equipment (that is, aircraft fault report(s) record) and the flight deck effect (FDE). These failures and warning messages are generated at different flight phases (take-off, cruise, and landing). The second database is the logs of aircraft maintenance activities (the comprehensive description of all aircraft maintenances recorded over time). The dataset is obtained from a fleet of aircraft. There are two families of aircraft in the long-range (A330) and the short aisle aircraft (A320). In each family, we target three components or functional items that are replaced due to unscheduled maintenance and study their failure behaviours. The behavioural patterns are then used to build a predictive model to predict their replacement. Data from the year 2011 to 2016 is used for training, while the reaming from 2018 to 2018 is used for testing.

Therefore, using the extremely imbalanced aircraft CMS dataset. We choose one approach from each of the categories of methods of handling imbalanced classification problems. SMOTE is chosen from the data-level approach, and the cost-sensitive ensemble method is chosen from the algorithm level approaches. We also evaluated the performance of our approach against the basic deep learning approaches for time-series predictions.

6.3.5.1 Parameter Setting

Discount factor γ = 0.5

The loss functions considered are;

Binary classification loss functions: Binary cross-entropy, Focal loss, Hinge loss.

Multiclass classification loss functions: Kullback Leibler divergence loss and sparse cross-entropy.

Proposed loss function: Re-scale weighted cross-entropy loss

The targeted aircraft Line-replaceable Unit (LRU): Line-replaceable unit is a component of an aircraft that is designed to be replaced easily at an aircraft line maintenance location in the event of failure [35]. We choose LRU of high practical value with an adequate number of known failure cases, identified by Functional Item Number (FIN). A330 –aircraft family: 4000KS - Electronic engine unit, 4000HA - High-pressure bleed valve, 5RV1 – Satellite data unit. A320- aircraft family: 11HB- Flow control valve, 10HQ - Avionics equipment ventilation computer, 1TX1 – Air traffic service unit.

6.3.5.3 Experimental Results

I. Comparing the proposed loss against Binary classification loss functions.

6.3.6 General Discussion on the Proposed Loss Function Against the Existing Binary and Multiclass Classification Loss Functions

We have proposed a loss function by re-scaling cross-entropy loss to handle the majority class's overwhelming gradient against the minority class. The proposed loss is expected to improve model performance in predicting rare component failure in complex systems. Figures 6-25, 6-26, 6-27, 6-28 and 6-29 present the comparative performance of the existing loss functions, that is, binary cross-entropy loss, focal loss, hinge loss, sparse cross-entropy and Kullback eligible divergence loss, respectively, with varying imbalance ratios. The proposed weighted re-scaled loss function is shown in Figure 6-30. We tested the loss functions via artificial imbalanced datasets using LSTM networks. Our target is to inform the best loss function to be used to classify an extreme imbalanced dataset. Therefore, in the experiment, varying imbalanced ratios (IR) were considered. We consider three cases (50%IR as balanced data, 20%IR and 5%IR imbalanced). In the case of 50%IR, since the data is not imbalanced, it shows that most classification methods can work well with this case. In the case of 20%IR, some of the imbalanced classification methods are cable of handling it; hence performance improvement is understood for certain types of data in a sequence to sequence learning. However, it was not clear whether training extreme imbalanced of less than 5%IR, numerical, time-series data using re-scale weighted loss approach would improve model performance as there might not exist some definite temporal contexts and observable in-text sequence learning.

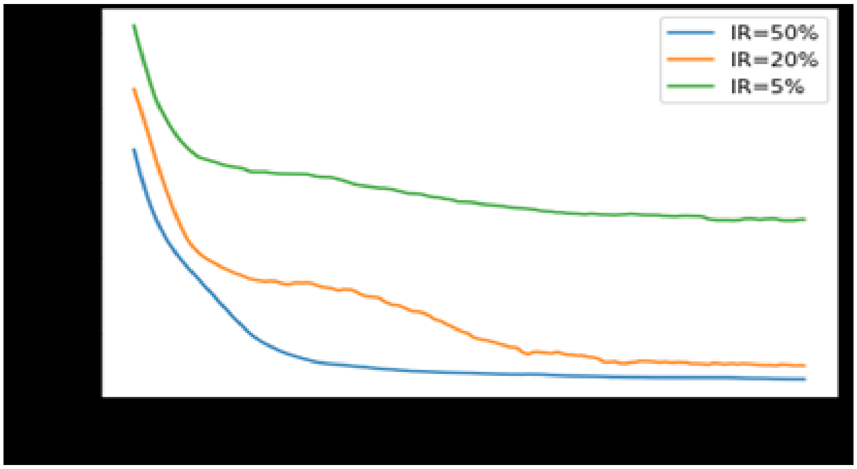


Figure 6- 24 Training the LSTM network with different imbalance ratio using focal loss

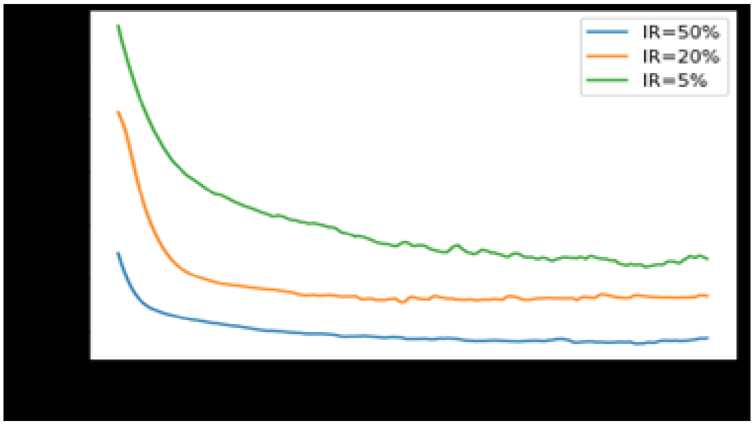


Figure 6- 25 Training the LSTM network with different imbalance ration using normal binary cross-entropy loss

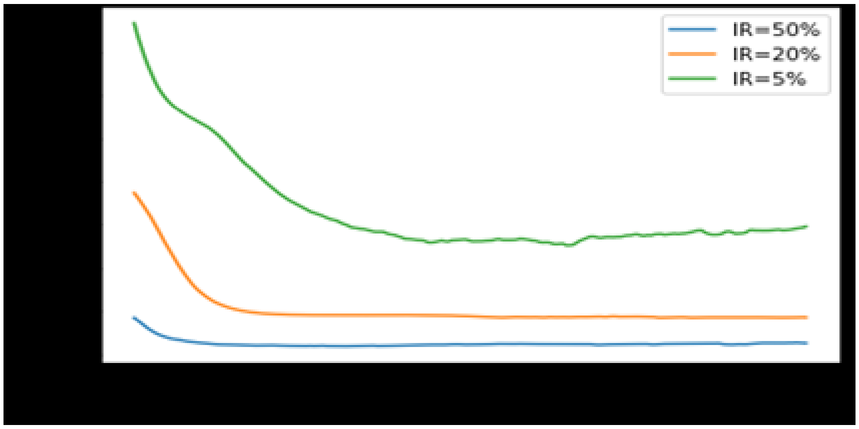


Figure 6- 26 Training the LSTM network with different imbalance ration using hinge loss

II. Comparing the proposed loss against multiclass-classification loss functions.

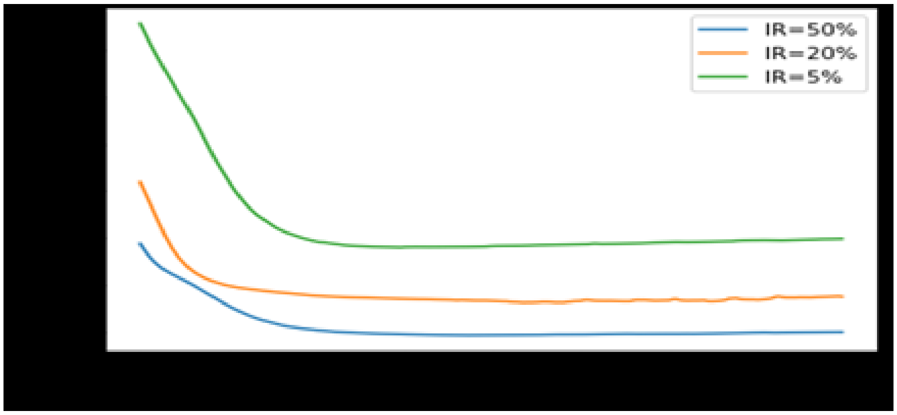


Figure 6- 27 Training the LSTM network with different imbalance ration using sparse cross-entropy

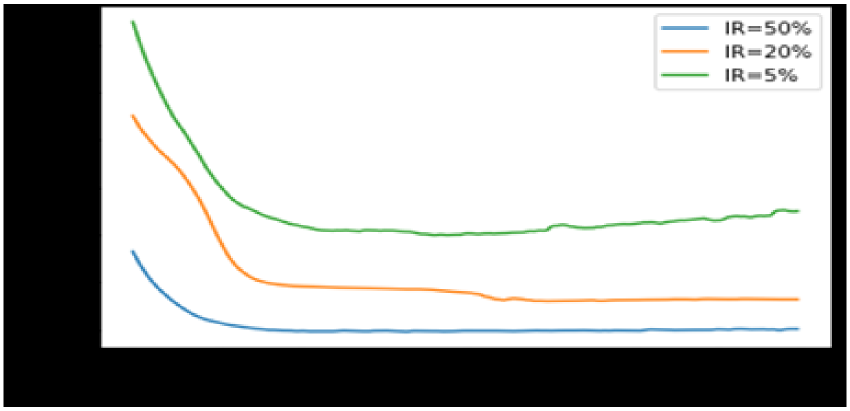


Figure 6- 28 Training the LSTM network with different imbalance ration using Kullback Leibler divergence loss

III. Proposed loss function: Re-scale weighted cross-entropy loss.

As observed in Figure 6-25 to 6-29, the balanced case that is 50%IR, the overall loss is approximately 0.1 in all the considered existing loss functions. Compared with the proposed method in Figure 6-30, the result is that as the training epoch size increases, the loss decreases rapidly, approaching zero. Observing the case of the imbalanced ratio of 20%IR, using the binary cross-entropy in Figure 6-25, it converges with an error of 0.21, focal loss in Figure 6-26 converges with an error of 0.1, hinge loss in Figure 6-27 with an error of 0.2, sparse cross-entropy in Figure 6-28 with an error of 0.21. Kullback Leibler divergence loss in Figure 6-29 with an error of 0.21. Comparing the case of 20% imbalanced ration performance with the proposed method in Figure 6-30 shows that it converges with an error of 0.05, which is far lesser than the closes focal loss with an error of 0.1, which indicates better performance.

Considering the extreme case where the imbalance ratio is 5%IR between classes, the results show that the existing loss function with worse error is the Kullback Leibler divergence loss in Figure 6-29, which converges with an error of 0.36. Simultaneously, the best is the focal loss function in Figure 6-28, with an error of 0.28. To compare with our approach in Figure 6-30, which converges with an error of 0.15. This indicates that our approach is feasible and efficient, achieving high performance and robustness via extreme imbalanced datasets. In summary, re-scaling weighted cross-entropy loss performs better than the existing binary and multiclass loss functions. Figure 6-

31 shows the overall model performance, and it achieved more than 80% accuracy on training and over 70% on the testing dataset.

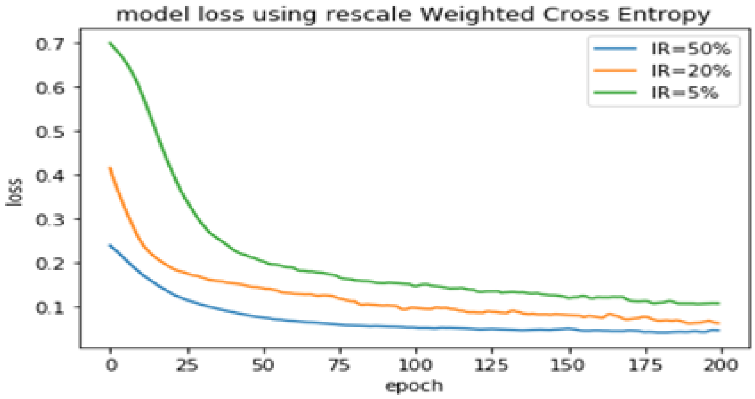


Figure 6- 29 Training the LSTM network with different imbalance ration using re-scaled loss function for imbalanced classification

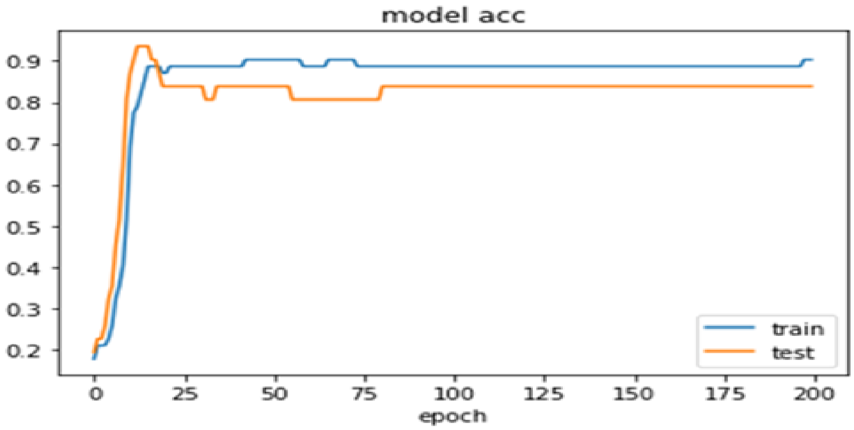


Figure 6- 30 Overall model performance using Rescaled –LSTM networks

6.3.6.1 Performance of the proposed method on aircraft central maintenance dataset.

We have proposed a re-scale LSTM network model that models the extreme imbalanced dataset to predict aircraft component removal. The challenge in predicting extreme rare component failure (which make up the imbalanced classification problem) is during model training, the overall total error cost representing the majority samples to have a negative impact on the minority class samples because most of the losses from the majority class will dominate the gradient, hence producing low-performance model. Therefore, we attempt to solve that problem by using a re-scale weighted loss function to control the overwhelming gradient from the majority class. Hence, producing better model accuracy.

Table 6-4 shows the results of our proposed approach compared with the state-of-the-art approach of handling imbalanced datasets. For comparison, we consider SMOTE from the data level approach and the ensemble cost-sensitive method from the algorithm level approach. The dataset used in this study is obtained from a fleet of aircraft; in the fleet, there are two aircraft families in the long-range (A330) and the short aisle aircraft (A320). In each family, we target three components that are replaced due to an unplanned breakdown. We study their failure behaviours and then use those behavioural patterns to build a predictive model to predict their future replacement. Modelling the problem as a binary classification, we predict each component replacement separately. The imbalanced ratio is extreme in each case because such replacements are rare in the dataset. We evaluate the model using precision, recall and the Geometric Mean. The choice of performance matrices is due to their effectiveness in evaluating imbalanced classification models.

Table 6- 4 Result based on performance comparison with state-of-the-art methods for imbalance learning

		SMOTE			**Ensemble + CS**			**R-LSTM**		
A330-aircraft family										
Comp.	***IR %***	***Precision***	***Recall***	***G-mean***	***Precision***	***Recall***	***G-mean***	***Precision***	***Recall***	***G-mean***
4000HA	0.47	0.87	0.65	0.74	0.88	0.75	0.80	**0.96**	**0.85**	**0.92**
4000KS	0.43	0.81	0.63	0.70	0.81	0.69	0.74	**0.93**	**0.89**	**0.85**
5RV1	0.44	0.80	0.64	0.71	0.83	0.70	0.76	**0.92**	**0.89**	**0.91**
A320- aircraft family										
11HB	0.28	0.75	0.65	0.69	0.82	0.73	0.77	**0.90**	**0.86**	**0.88**
10HQ	0.31	0.83	0.60	0.70	0.89	0.74	0.80	**0.92**	**0.87**	**0.89**
1TX1	0.64	0.88	0.66	0.76	0.91	0.85	0.87	**0.95**	**0.88**	**0.91**

It can be observed from the results that in each of the considered cases, the proposed model outperformed others in both G-mean score and recall. We can observe that even though the imbalance ratio for all cases considered is less than 1%, the model recall is more than 80%. This show the model robustness in handling extreme imbalanced classification problem. The G-mean is more than 80% in all cases, showing that the model can better reduce the false-positive rate.

We also experimented with confirming that the proposed approach is superior to other deep learning approaches for time-series predictions. We modelled the problem as time series, binary classification problem. For this reason, we, first, transform the data into a suitable format for deep learning. The experimental results are evaluated on the log-based aircraft CMS dataset. The result shows that the proposed method performed better in predicting each LRU replacement.

Table 6- 5 Result based on performance comparison with basic deep learning approaches for time-series predictions

		RNN			**LSTM**			**R-LSTM**		
A330-aircraft family										
Comp.	***IR %***	***Precision***	***Recall***	***G-mean***	***Precision***	***Recall***	***G-mean***	***Precision***	***Recall***	***G-mean***
4000HA	0.47	0.80	0.61	0.69	0.83	0.61	0.71	0.96	0.85	0.92
4000KS	0.43	0.68	0.51	0.57	0.81	0.59	0.68	0.93	0.89	0.85
5RV1	0.44	0.68	0.52	0.58	0.82	0.60	0.70	0.92	0.89	0.91
A320- aircraft family										
11HB	0.28	0.70	0.51	0.59	0.69	0.53	0.59	0.90	0.86	0.88
10HQ	0.31	0.78	0.57	0.66	0.72	0.58	0.64	0.92	0.87	0.89
1TX1	0.64	0.82	0.60	0.70	0.80	0.66	0.72	0.95	0.88	0.91

Table 6-5 summarises the experiment results comparing the existing methods (RNN and normal LSTM) to the proposed rescaled-LSTM networks. The result depicts that the proposed method is superior in both recall and precision. This indicates that its suitability for a time series prediction.

6.3.7 Conclusion and Future Work

This paper has identified an imbalanced dataset as the main challenge for performance degradation in developing aircraft predictive maintenance models. A proposed re-scale loss function has been introduced into the LSTM networks to focus on hard-to-learn examples from the minority class. This will address the class imbalance problem in aircraft component replacement predictive models. The experimental result indicates that R-LSTM has a better performance than other similar imbalance learning techniques. We also achieved a significant level of improvement in the reduction of false-positive and false-negative rates. In the future, we hope to develop this work further by looking at the effect of class overlapping in the process of over-sampling the minority class in the imbalanced learning context. We will also look at improving model performance by analysing the model internal structure to predict more accurately component replacement in the desired time window in advance -before failure to carry out actionable maintenance.

6.3.8 References

1. Eickmeyer J., Li P., Givehchi O., Pethig F., Niggemann O. Data Driven Modeling for System-Level Condition Monitoring on Wind Power Plants. Int. Work. Princ. Diagnosis. 2015; 1507: 43–50.

2. Ossai C. Integrated Big Data Analytics Technique for Real-Time Prognostics, Fault Detection and Identification for Complex Systems. Infrastructures. 2017; 2(4): 20. Available at: DOI:10.3390/infrastructures2040020

3. Wagner C., Saalmann P., Hellingrath B. Machine Condition Monitoring and Fault Diagnostics with Imbalanced Data Sets based on the KDD Process. IFAC-PapersOnLine. Elsevier B.V.; 2016; 49(30): 296–301. Available at: DOI:10.1016/j.ifacol.2016.11.151

4. Yusof R., Kasmiran KA., Mustapha A., Mustapha N., Zin NAM. Techniques for handling imbalanced datasets when producing classifier models. J. Theor. Appl. Inf. Technol. 2017; 95(7): 1425–1440.

5. Douzas G., Bacao F. Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Systems with Applications. Elsevier Ltd; 2018; 91: 464–471. Available at: DOI:10.1016/j.eswa.2017.09.030

6. He H. Imbalanced Learning. Self-Adaptive Systems for Machine Intelligence. New Jersey: John Wiley & Sons, Inc.,Hoboken, New Jersey.; 2011. 44–107 p. Available at: DOI:10.1002/9781118025604.ch3

7. Branco P., Torgo L., Ribeiro RP. A Survey of Predictive Modeling on Imbalanced Domains. ACM Computing Surveys. 2016; 49(2): 1–50. Available at: DOI:10.1145/2907070

8. Nghiem LT., Thu TT., Nghiem TT. MASI: Moving to adaptive samples in imbalanced credit card dataset for classification. 2018 IEEE International Conference on Innovative Research and Development, ICIRD 2018. 2018. pp. 1–5. Available at: DOI:10.1109/ICIRD.2018.8376315

9. Sajana T., Narasingarao MR. A comparative study on imbalanced malaria disease diagnosis using machine learning techniques. Journal of Advanced Research in Dynamical and Control

Systems. 2018; 10: 552–561. Available at: DOI:https://www.jardcs.org/backissues/abstract.php?archiveid=2962&action=fulltext&uri=/backissues/abstract.php?archiveid=2962

10. Jiao Z., Jia G., Cai Y. A new approach to oil spill detection that combines deep learning with unmanned aerial vehicles. Computers and Industrial Engineering. Elsevier; 2018; (September): 1–12. Available at: DOI:10.1016/j.cie.2018.11.008

11. Fernández Alberto, Garcia Salvador, Galar Mikel, Prati Ronaldo, Krawczyk Bartosz HF. Learning From Imbalanced Data Sets. 2018. Available at: DOI:https://link.springer.com/content/pdf/10.1007%2F978-3-319-98074-4.pdf (Accessed: 6 May 2019)

12. Haixiang G., Yijing L., Shang J., Mingyun G., Yuanyue H., Bing G. Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications. 2017; 73: 220–239. Available at: DOI:10.1016/j.eswa.2016.12.035

13. Abd Elrahman SM., Abraham A. A Review of Class Imbalance Problem. Journal of Network and Innovative Computing. 2013. Available at: DOI:www.mirlabs.net/jnic/index.html (Accessed: 23 January 2019)

14. Hochreiter S., Urgen Schmidhuber J. Lstm. Neural Computation. 1997; 9(8): 1735–1780. Available at: DOI:10.1162/neco.1997.9.8.1735

15. Lin TY., Goyal P., Girshick R., He K., Dollar P. Focal Loss for Dense Object Detection. Proceedings of the IEEE International Conference on Computer Vision. 1993; 2017-Octob: 2999–3007. Available at: DOI:10.1109/ICCV.2017.324

16. Kukar M., Kononenko I. Cost-sensitive learning with neural networks. 13th European Conference on Artificial Intelligence. 1998; : 445–449. Available at: http://pdf.aminer.org/000/165/499/cost_sensitive_learning_with_neural_networks.pdf

17. Zhou Z., Member S., Liu X. Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem. 2006; 18(1): 1–14.

18. Johnson JM., Khoshgoftaar TM. Survey on deep learning with class imbalance. Journal of

Big Data. Springer International Publishing; 2019; 6(1). Available at: DOI:10.1186/s40537-019-0192-5

19. Buda M., Maki A., Mazurowski MA. A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks. 2018; 106: 249–259. Available at: DOI:10.1016/j.neunet.2018.07.011

20. Lin M., Tang K., Yao X. Dynamic sampling approach to training neural networks for multiclass imbalance classification. IEEE Transactions on Neural Networks and Learning Systems. IEEE; 2013; 24(4): 647–660. Available at: DOI:10.1109/TNNLS.2012.2228231

21. Khan SH., Hayat M., Bennamoun M., Sohel FA., Togneri R. Cost-sensitive learning of deep feature representations from imbalanced data. IEEE Transactions on Neural Networks and Learning Systems. 2018; 29(8): 3573–3587. Available at: DOI:10.1109/TNNLS.2017.2732482

22. Tallón-ballesteros JDAJ., Hutchison D. Data Engineering and Automated Learning – IDEAL 2017. 2017.

23. Lee H., Park M., Kim J. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. Proceedings - International Conference on Image Processing, ICIP. IEEE; 2016; 2016-Augus: 3713–3717. Available at: DOI:10.1109/ICIP.2016.7533053

24. Mas MI., Fanany MI., Devin T., Sutawika LA. An initial exploration of the suitability of long-short-Term-memory networks for multiple site fatigue damage prediction on aircraft lap joints. 2017 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2017. 2018; 2018-Janua: 415–421. Available at: DOI:10.1109/ICACSIS.2017.8355067

25. Wu Z., Guo Y., Lin W., Yu S., Ji Y. A weighted deep representation learning model for imbalanced fault diagnosis in cyber-physical systems. Sensors (Switzerland). 2018; 18(4). Available at: DOI:10.3390/s18041096

26. Arora V., Sun M., Wang C. Deep Embeddings for Rare Audio Event Detection with Imbalanced Data. ICASSP, IEEE International Conference on Acoustics, Speech and Signal

Processing - Proceedings. 2019; 2019-May: 3297–3301. Available at: DOI:10.1109/ICASSP.2019.8682395

27. Shen Z., Zhang Y., Lu J., Xu J., Xiao G. A novel time series forecasting model with deep learning. Neurocomputing. Elsevier B.V.; 2019; (xxxx). Available at: DOI:10.1016/j.neucom.2018.12.084

28. Yao S., Zhao Y., Shao H., Zhang A., Zhang C., Li S., et al. RDeepSense. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies. 2018; 1(4): 1–26. Available at: DOI:10.1145/3161181

29. Yeo I., Balachandran K. Sentiment Analysis on Time-Series Data Using Weight Priority Method on Deep Learning. 2019 International Conference on Data Science and Communication (IconDSC). IEEE; 2019; : 1–7. Available at: DOI:10.1109/icondsc.2019.8816985

30. Gamboa JCB. Deep Learning for Time-Series Analysis. 2017; Available at: http://arxiv.org/abs/1701.01887

31. Jordan M., Kleinberg J. Cross Entropy Optimization. Pattern Recognition. 2006. 791–799 p. Available at: DOI:10.1641/B580519

32. Rubinstein RY., Kroese DP. SIMULATION AND THE Third Edition. 2017.

33. Popkes A. Kullback-Leibler Divergence Interpreting the KL divergence. 2019; : 1–7.

34. Tripathi S., Kumar A., Ramesh A., Singh C. Focal Loss based Residual Convolutional Neural Network for Speech Emotion Recognition. arXiv. 2019; : 1–10.

35. Maschinenbau F. Specification and Evaluation of Prediction Concepts in Aircraft Maintenance. 2017;

CHAPTER 7: Application of Deep Reinforcement Learning for Extremely Rare Failure Prediction in Aircraft Maintenance

This chapter presents the implementation of deep reinforcement learning for the classification of an imbalanced dataset. In this approach, the problem is formulated as a Markov-decision process framework and solved using the deep reinforcement learning (e.g. deep Q-learning networks) techniques.

The use of aircraft operational logs to predict potential failure that may lead to disruption poses many challenges and has yet to be fully explored. Given that aircraft are high-integrity assets, failures are extremely rare, and hence the distribution of relevant data containing prior indicators will be highly skewed to the normal (healthy) case. This will present a significant challenge in using data-driven techniques because the model will be biased to the heavily weighted no-fault outcomes. This paper presents a novel approach for predicting unscheduled aircraft maintenance action based on deep reinforcement learning techniques. The algorithm transforms the rare failure prediction problem into a sequential decision-making process optimised using a reward system that penalises proposed predictions that result in a false diagnosis and preferentially favours predictions that result in the right diagnosis. A log data from the aircraft central maintenance system is used for model validation; the data is directly associated with the physical health aspects of the aircraft components. The influence of extremely rare failure prediction on the proposed method is analyzed. The new approach was compared with existing cost-sensitive and oversampling methods, and performance was evaluated based on G-mean and false-positives rates. The proposed approach shows the superior performance of 20.3% improvement in G-mean and 97% reduction in false-positive rate.

7.1 Introduction

In recent times, the concept of predictive maintenance has continued to advance, especially in a complex system such as an aircraft. Predictive maintenance is designed to monitor in-service equipment's health condition and forecast maintenance needs. It provides a cost-benefit compared to time-based approaches such as preventive maintenance because maintenance is carried out only when needed [1]. As the popularity of predictive maintenance models increases in the aviation industry, one of the critical challenges is dealing with unplanned failures, i.e. rarely reported events. In other words, the challenge of learning from an extremely imbalanced dataset using standard machine learning algorithms.

Furthermore, using the data from operational equipment logs to develop predictive models poses many challenges that have not yet been fully explored, as logs are mainly used for anomaly detection and debugging failure. The logs generated in complex systems such as aircraft are mostly multivariate time series (multiple interrelated streams of data are recorded simultaneously). This type of data is commonly recorded from several monitoring systems, such as the condition-based or sensors, collected over time. They may, therefore, be regarded as complex multivariate time-series data. Given that aircraft are high-integrity assets, failures are extremely rare, and hence the distribution of relevant data containing prior indicators will be highly skewed to the normal (healthy) case. This will present a significant challenge in using data-driven techniques to 'learning' relationships/patterns that depict fault scenarios since the model will be biased to the heavily weighted no-fault outcomes.

Some of the characteristics of a system log that cause a challenge in predictive modelling are:

(i) Heterogeneous in nature containing symbolic sequences, numeric time-series, categorical variables and unstructured text. It is a non-trivial task to translate free-text log messages into meaningful features.

(ii) System log volume can be large in complex systems, which poses computational challenges.

(iii) Having a rare occurrence of failure results in a lack of enough information to anticipate certain specific families of faults.

Thus, this study investigates the use of aircraft operational log-based data to develop a predictive model for rare failure prediction in aircraft. Also, to determine which variables are likely to indicate the target failures. An issue of predictive maintenance lies in the rigid nature of data (data changing over time). If correct parameters are not built-in, it can risk incorrect forecasts and erroneous ‘fault’ messages. For instance, based on historical behaviour, if a maintenance operator forecasts that a component will fail within 100 flights, they might schedule removal to prevent operation failure. However, upon removal, the part may test as no fault found (NFF), costing the operator unnecessary time and money. Therefore, developing a robust predictive model is necessary, especially for safety-critical equipment such as aircraft.

In order to make use of log-based data to develop a robust predictive maintenance model, generally, the first step is to interpret the logs, filter out a large amount of noise (that is, data irrelevant to the set goal) and extract predictive features. Also, the known failure cases need to be collected for learning and evaluation. The problem needs to be transformed into an appropriate learning scenario, and a performance measure that reflects real-world needs must be determined. Figure 7-1 shows the proposed process of discovering knowledge from raw data. The raw heterogeneous and multivariate data collected from different sources is stored in a database. The raw data usually contains many analytical challenges requiring pre-processing, such as data incompleteness, lack of example behaviours and trends, missing or null values, lack of exact features of interest, and noise. Data pre-processing and transformation (into a suitable format for machine learning) occurs in Stage 2 of Figure 7-1. A feature engineering (FE) process is carried out at stage three; it helps collect relevant features related to the desired goal. FE is the integral and critical step of the machine learning process because the quality of data and the right features contribute majorly to a predictive model’s performance. After the pre-processing and FE phase, the data is divided into training and validation. Stage four is where the machine learning algorithm for pattern recognition or classification is trained using the training data. The model is then evaluated at stage five. The outcome can then give insightful knowledge for more informed decision making.

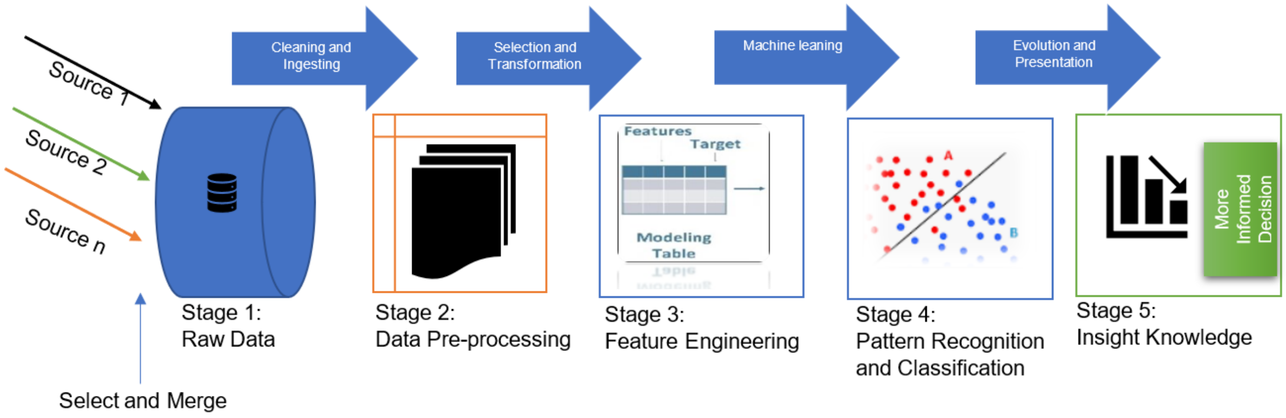


Figure 7- 1 Basic Data Knowledge Discovery Process

The pre-processed dataset usually has a skewed distribution in a rare failure prediction problem. For example, in the ACMS dataset, the non-failure represent negatively labelled samples, and the failure represent positively labelled samples. The negative samples far outnumber the positively labelled, causing the data to be highly imbalanced. The disproportion between classes can be very low (e.g. 5% or less). Various solutions for the slight rare failure problem (say proportions of 40:60 to 30:70) have been suggested in the literature. However, in a situation where the imbalance ratio is extreme, say less than or equal to 5%, the problem becomes more challenging to handle [2][3]. In such a scenario, the standard approaches for normal failure prediction (such as statistical approaches, traditional machine learning algorithms and associated rules) become limited [4–6]. The reason is that most normal failure patterns are similar to each other and are substantially represented.

In contrast, rare failure is typically one-of-a-kind, and hence it becomes difficult to learn temporal patterns using traditional machine learning approaches. That is why many aircraft predictive maintenance models are based on simple “threshold” monitoring rules capable of detecting only simple faults and, consequently, having high false-positive rates (FPR) [7]. Hence, it is vital to provide an accurate prediction of failures and, at the same time, have a very low FPR. That can improve the effectiveness of the aircraft health monitoring systems and, in turn, enhance the availability of the aircraft.

This study considers the case of developing a model to predict unplanned failure and replacement of aircraft components. The dataset used contains extremely rare failures of the target component. The imbalance ratio for each target component is less than 3% of the total dataset, making it difficult to develop a predictive model effectively using the existing traditional machine learning approaches. Therefore, this study aims to show the applicability of deep reinforcement learning for training an extremely rare failure predictive model instead of the widely used machine learning or deep learning methods for slightly rare failure predictions. The proposed model is trained using a real-world aircraft central maintenance system (ACMS) dataset.

The proposed approach considers the problem of extremely rare event prediction from a reinforcement learning point of view. The problem is formulated as a Markov sequential decision-making process and solved by combining reinforcement learning with deep neural networks. The approach enables the model to remember a long sequence of failure patterns. The reward function is specifically constructed to counter agent bias towards the majority class during model training. Figure 7-2 shows the interaction between the elements of reinforcement learning. Here, the agent-classifier takes action in an environment; transition through the time series ACMS dataset is considered an environment in the proposed approach. A reward is returned based on the action taken (classify pattern as fault or non-fault) at a given state.

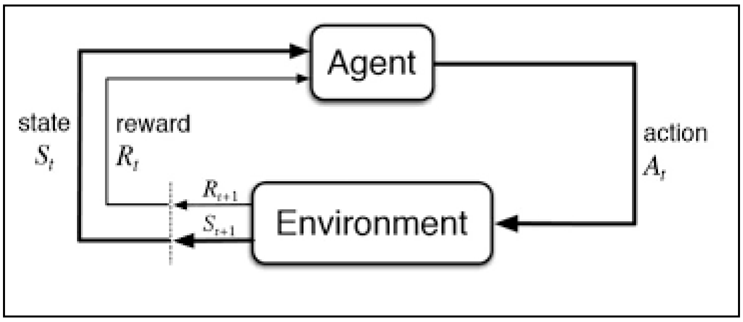


Figure 7- 2 Visual representation of iterative feedback loop of actions, states, and rewards in reinforcement learning

Rationale: DRL algorithms were traditionally designed for performance optimisation with very large input space [8]. Therefore, exploring the application of DRL approaches for complex systems large log-based datasets can significantly benefit the predictive maintenance, especially that data is continually increasing in dimension [9]. The rationale for the proposed method is to explore the applicability of deep reinforcement learning for extremely rare prediction problems, purposely for performance optimisation in complex systems predictive maintenance models, to minimize downtime and increase the utilization rate of the vehicles or components. The motivation for the possible performance improvement in the proposed algorithm is the combination of the convolutions in deep neural networks that enhance learning relationships between variables in the dataset. Also, the reward function, which helps to counter bias during model training and prioritised experience replay memory, which instead of uniformly sampling transactions from replay memory, employs a prioritised approach that also entails replaying the important transactions more frequently. Hence, optimising the learning process. Also, DRL uses a reward function to optimise future rewards, in contrast to a machine learning (regression or classification) model that predicts the probability of future outcomes. Therefore, it can be assumed that deep reinforcement learning methods are ideally best for imbalanced classification problems because of its learning mechanism and specific learning environment and reward function.

This paper presents a novel approach using deep reinforcement learning techniques to predict unplanned aircraft maintenance actions using data from operational flight logs and maintenance report information. The approach first identifies relevant temporal patterns that correspond to each component failure. It then transforms the problem into a sequential decision-making process that is optimized using deep reinforcement learning algorithms utilizing a reward system that penalizes proposed predictions leading to a false diagnosis and preferentially favours predictions that lead to a correct diagnosis. The failure messages in the ACMS data is directly associated with physical health aspects of the vehicle, asset or component (such as pressure, vibration, temperature, acoustics, viscosity, flow rate data). The patterns that are input to the algorithm represent the history state of the components, and they are labelled as failure or non-failures. The reward function is specifically constructed to counter agent bias towards the majority class during model training. The strategy allows adequate handling of extremely imbalanced problems in predictive maintenance modelling. The influence of extremely rare failure prediction on the proposed deep reinforcement learning models is analyzed.

The main contributions of this paper are as follows:

1. To show a novel application of deep reinforcement learning to predict extremely rare failure problems in complex aircraft systems. The new deep reinforcement learning approach is designed to capture the patterns of extremely rare component failures adequately. The model is trained to predict aircraft component replacement well in advance of failure. The technique includes designing and developing an environment for the state-action, a reward

function for rewarding agent-classifier actions, and the unique arrangement of a deep neural network architecture for policy optimization.

2. The new method is validated using a real-world aircraft central maintenance system dataset. Exploring the ACMS dataset for developing a predictive maintenance model is a significant contribution because of its heterogeneous nature, challenging to analyze.

The rest of this paper is organized as follows. Section 7.2 provides related work. Section 7.3 presents the proposed new method and its implementation. Section 7.4 presents the case study. Section 7.5 shows the results and discussion, and the conclusion is presented in section 7.6.

7.2 Related Work

One of the design goals of predictive maintenance is to avoid unexpected failures by monitoring the vehicle condition and providing failure alerts well in advance. Predictive maintenance models are developed to forecast when likely the vehicle will fail, so that maintenance can be systematically scheduled to occur way in advance before the failure point. Predictive maintenance can be modelled in physics-based, knowledge-based, and data-driven-based [10]. Physics-based modelling can be defined as a simplified mathematical description of a system or process to assist calculations and predictions [11]. The prediction is based on a mathematical equation inside the mode; therefore, it uses a limited amount of data compared to other methods. However, the physics-based model is challenging to create and implement, especially for complex systems, because it is sensitive to the system's design and material properties. Also, enough component information and a good knowledge of the failure mechanism is highly required to formulate the model.

The knowledge-based model, also known as the expert system, uses defined rules or fuzzy logic to solve complex problems. The rules are set based on the knowledge of a domain expert. Converting domain knowledge to a set of rules is challenging, requiring another prognostics technique. Also, the set of rules needs to be updated anytime there is any system update. This process can be cumbersome and sometimes impractical, especially in a complex system with many components and processes.

The data-driven approach involves training machine learning algorithms using large historical datasets to learn a system behaviour model automatically. A data-driven approach is easy to implement, flexible, adaptable with a low cost of implementation. However, large historical data representing failure is needed, and getting such data is always challenging. However, the advancement in technology data is increasingly available, making it more appealing to use a data-driven approach for developing predictive maintenance models in complex systems. To the optimised performance of predictive modelling, the hybrid of the two or three approaches can be explored [12], which is one of the focuses of this study.

7.2.1 Rare Event prediction

The challenge of predicting rare events has been around for some time and is still an ongoing research area [1]. Many solutions have been proposed in the literature, especially related to the maintenance of heavy industrial equipment and other domains that require rare event prediction. The existing solutions are primarily found in statistical methods and machine learning methods. Examples of statistical methods are the extreme value theory or extreme value analysis (EVA) [13,14] and the peak over threshold (POT) methods [15]. These methods deal with extreme deviation from the mean of a probability distribution in a dataset [16,17]. Statistical methods draw population inferences from a sample, whereas machine learning finds generalizable predictive patterns [18]. Machine learning approaches are desirable in this study because they are particularly helpful when harnessing knowledge from large heterogeneous datasets. They are more effective and efficient compared to other data mining and analysis methods.

Machine learning approaches are divided basically into supervised, unsupervised, and reinforcement learning. Other hybrid learnings are semi-supervised, self-supervised and multi-instance learning. Supervised learning techniques involve learning or inferring using labelled training datasets. An example of supervised learning is seen in building a model for rare event prediction based on labelled data (the training set) [19]. One of the strongest advantages of supervised methods is that they can easily be validated, but the training data must be labelled.

On the other hand, unsupervised learning involves developing models using unlabelled datasets; this is mainly used for problems such as anomaly detection, deviation detection, outlier analysis, and exception mining. These methods analyse each event one after another to determine how similar or dissimilar they are to the majority. Their success depends on the choice of parameters, such as similarity measures and dimension weighting. Therefore, because the dataset used in this study has defined labels, the supervised machine learning approach is considered.

Furthermore, rare event prediction can also be modelled using association rules (knowledge-based). However, this approach is more effective for a small and simple system [20], not the large heterogeneous datasets studied here. The use of associative rules for a large and complex system is quite challenging and, in some cases, impractical because domain experts need to continually update the rules in the event of any upgrades or changes, which is time-consuming and cumbersome [21][22]. Another potential approach is reinforcement learning which can be considered from a sequential learning point of view. In this type of learning, an agent takes the best actions sequentially in a particular environment in order to maximise cumulative rewards [23]. The current study focuses on the deep reinforcement learning approach.

Why is deep reinforcement learning considered for extremely rare event prediction instead of the standard deep learning or machine learning approach? It is a legitimate question, and the answer is subjective. Existing machine learning algorithms can handle the data imbalance problem in diverse dimensions depending on the type of dataset. However, considering that a situation where the target events are extremely rare, those methods become limited [3,4,24]. For instance, an imbalanced classification problem can be handled at the data level either by under-sampling the majority (negatively labelled) samples to balance with the minority class (positively labelled) or over-sample the minority class by creating more synthetic samples. Then the model can be trained using any existing machine learning algorithm. In this case of under-sampling, if the imbalance ratio is say1:200, in a total of a million records, about 0.5% will remain in the positively labelled dataset. After under-sampling, a total of approximately 1% of the original dataset will be left. The standard machine learning algorithms (such as Support Vector Machine, Decision Tree or Random Forest) can be used to train the model with data of this size. However, the potential information in the remaining ~99% of data left out will not be utilized, producing a low-sensitivity model [25].

Another approach could be to over-sample the minority class, then use machine learning to train the model. This approach has the drawback of increasing the likelihood of overfitting since it replicates the minority class examples. The Synthetic Minority Oversampling Technique (SMOTE) [26] has been developed to mitigate overfitting in random oversampling by taking a subset of data from the minority class as an example and then creating new synthetic similar instances. However, SMOTE has the drawback of not considering neighbouring examples from other classes when generating synthetic samples. That can cause overlapping of classes and can also introduce additional noise into the training data. SMOTE is also ineffective in high dimensional data, as argued by Lusa et al. [27]. In recent times, many solutions have been proposed to correct the drawbacks of SMOTE[28–

30] and other novel solutions which are specific to either the application domain or dataset in question, as presented by Alberto et al. [25].

Furthermore, another approach is to transform the dataset and then uses deep learning methods to train the model. Recent examples of time-series-based deep learning models have been proved to provide state-of-the-art performance in handling slightly rare event prediction problems. For example, the combination of an Auto-encoder with LSTM or GRU deep neural networks has been shown in Maren et al. [31] and Di et al.[32]. Although these models have continued to improve over time, the challenge of handling an extremely imbalanced dataset, or extremely rare event prediction, remains an area that requires continuous improvement. For instance, model performance degradation is seen in training deep neural networks with an imbalanced dataset. Deep learning methods are affected by a highly imbalanced dataset because the overall total error cost representing the majority samples impacts the minority class samples by overwhelming the gradient responsible for updating the model's weights. Hence creating a biased model that will produce a high FPR [31,33]. Therefore, the open literature lacks a unified solution to handling extreme imbalance classification problems, especially for large heterogeneous ACMS datasets. Hence, this study seeks to provide a solution to an extreme imbalance problem using a deep reinforcement learning (DRL) approach. The solution aims to optimise the data-driven model's performance by avoiding biases and reducing the false positive rate.

7.2.2 Deep reinforcement learning for predictive model

The integration of deep learning with reinforcement learning, known as DRL, to optimise model performance is gaining more research attention, and it is producing state-of-the-art solutions [34]. For instance, the integration of deep learning and reinforcement learning has led to the emergence of a novel technique called the deep Q-network (DQN)[3,23,35]. DRL has made the application of reinforcement learning attractive in different domains. One such domain is in developing predictive maintenance models for complex systems. A detailed survey on deep reinforcement learning and its applications can be found in a study by Kia et al. [23]. The DRL application can be seen in robotics and gaming [30][31], where different techniques are used to achieve the desired results. Also, in communication and networking [36], detecting and predicting failure notes in the network and cyber security[37] for detecting fraudulent events in the system. In the financial sector, DRL is used for solving complex business problems [38][39] and for inventory management and resource allocation [40]. Others are in medicine [41], engineering and manufacturing [42][43].

Recently, the application of DRL for equipment maintenance is gaining more research attention. A study by Knowles et al. [44] has shown how to integrate reinforcement learning into condition-based maintenance. Rocchetta et al. [45] developed a framework based on DQN to optimise power grid equipment's operation and maintenance. Both approaches are based on Markov Decision Process (MDP) and DRL. The applicability of deep reinforcement learning for equipment health indicator learning is also shown in a study by Chi Zhang et al. [46]. However, the open literature lacks any exhaustive study that shows how extremely rare event prediction in complex systems can be modelled using deep reinforcement learning approaches, which our study seeks to fill.

The current study is motivated by the fact that exploring the application of DRL methods for real-world problems, such as rare equipment failure prediction, for potential performance optimization opportunities. In data classification problems, DRL has served better in removing noise from data and learning hard temporal features, improving predictive models' performance [16]. Lin et al. [3] pointed out that deep reinforcement learning methods are ideal for imbalanced classification problems because of their learning mechanism and specific training environment and the control of the learning process using reward function. DRL uses a reward function to optimize future rewards, in contrast to a machine learning (regression or classification) model that predicts future outcomes probability.

The DRL framework can be constructed by combining a deep neural network and reinforcement learning. That can be seen in Q(λ)-learning [47], where the reward function can give a high reward or a penalty for an action taken by the agent-classifier on a positively labelled class (minority). With more attention given to the minority class, the algorithm can respond favourably to both classes during learning, hence enhancing the resulting model's effectiveness.

As demonstrated by this review of the open literature, research on the application of deep reinforcement learning for extreme rare event prediction in complex systems is limited. Thus, this paper demonstrates the application of deep reinforcement learning in aircraft predictive maintenance modelling, focusing on developing a model to predict extremely rare failure using a heterogeneous log-based ACMS dataset.

7.3 Methodology

7.3.1 Description of reinforcement learning based on the Markov Decision Process

In reinforcement learning and Markov Decision Process (MDP), the agent interacts with an environment $\mathcal{E}$ sequentially over a discrete-time step t. The agent takes action a_t at time t after observing the state s_t. Based on the agent's action a_t, reward r_t is returned. The process can be represented as a 7-tuple of $\boldsymbol{M} = (\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{P}, \boldsymbol{R}, \boldsymbol{s_0}, \boldsymbol{\gamma}, \boldsymbol{T})$, where S is the set of states. A is the set of actions. P is the transition probability distribution represented as $(P{:}\,SAS \ \rightarrow R^{+})$. R is the reward function, represented as $R{:}\,SA \ \rightarrow R$ and R^{+} a returned immediate reward received after transitioning from state s to next state s', due to action a. s_0 is the initial state distribution defined as $s_0{:}\,S \ \rightarrow R^{+}$. γ is the discount factor $\gamma \in [0,1]$, a lower discount factor motivates the decision-maker to favour taking actions early rather than postponing them indefinitely. T is the transitional probability distribution.

Once the MDP is defined, the target is to have an agent that can determine, at state s_t, which best next action to take in order to maximize the reward r_t. A gradient descent function can be used to maximize the reward based on a defined policy π_θ. For example, the agent takes an action $\hat{y}_t \in A$ with respect to the optimal policy $\pi(\hat{y}_t|s_t){:}\ SA \ \rightarrow R^{+}$ and observed reward r_t for that action. The cumulative discount sum of the rewards is the objective function optimized by the policy π_θ. The optimal policy is created using a value function, which is a defined estimated value related to each state. The value function can either be a V-function [48], which estimates the value for each state, or the Q-function [48], which estimates the value for each pair of state-action $Q(s,a)$. The basic transaction of Q-learning keep a lookup table, in contrast to the deep Q-networks which leverages the use of replay memory to store trajectory transactions and the stored interaction are fetched from the replay memory in mini-batches to train the deep neural networks [8]. In other words, deep Q-learning fits the Q-function with deep neural networks.

MDP based models are used for planning future action and rewards. Methods of solving reinforcement problems based on planning are either model-based or model-free. The model-based technique is when transitional probability T and reward R are known. In this case, the optimization process can learn from T and R. The model-free approach is when T and R are unknown. In that case, the optimization process will directly learn the best policy without knowing T and R using trial-

and-errors learners [49]. In our implementation, we adapt a State Action Reward State Action (SARSA) learning and Deep Q-network (DQN) methods [50][47] which are based on a model-based reinforcement learning approach. SARSA is an on-policy model meaning the agent gets the optimal policy and uses it to act, while Q-learning is off-policy because it estimates the reward for future action and appends a value to the new state without using any greedy policy [50].

7.3.2 Formulation of Rare Failure Prediction Framework Based on Markov Decision Process

To formulate the DRL-based rare failure prediction approach using the log-based ACMS dataset. The problem is considered as a sequence-to-sequence learning process, where the agent serves as a classifier. The agent receives patterns proceeding with each failure sequentially and classifies each pattern as either failure or non-failure. The environment then returns a reward based on the agent's action. A positive reward is returned if the agent makes a correct classification; otherwise, a negative reward is returned. In the process, the agent will learn optimal behaviour from the environment and subsequently improve the agent classification accuracy.

Assume the training dataset is

$$D = \left\{\left[\left(x_{1,1}, x_{1,2} \dots x_{1,n}\right), (y_1)\right], \left[\left(\left(x_{2,1}, x_{2,2} \dots x_{2,n}\right), (y_2)\right)\right], \dots, \left[\left(\left(x_{\mathrm{m},1}, x_{\mathrm{m},2} \dots x_{\mathrm{m},n}\right), (y_{\mathrm{n}})\right)\right]\right\},$$

Where $x_{i,j}$ is the failure pattern and y_is the labels.

Table 7-1 shows the sample of the data and the interaction. The training dataset contains n-number of features and their corresponding labels. To transform the data for the DRL application pattern related to each target event with its corresponding labels is considered as state S. At every given state, the agent-classifier takes action by considering patterns related to each event as inputs and then performing a classification action a at time t. Based on the action taken, a reward r_t is returned. At the end of each trajectory, a cumulative reward R_t is returned, and the transaction is recorded in a replay buffer.

Table 7- 1 Representation of interaction of the agent with the environment

n-Features					Labels	Agent classifier
	x1	x2	x3	xn	yi	-
The pattern of event 1	St				at	← rt
The pattern of event 2	St+1				at+1	←rt+1
…	…				…	…
The pattern of event n	St+n				at+n	←Rn

A window is defined using the flight leg, and the end of each window is considered a trajectory. The agent-classifier can learn which action is favourable at a future given state by taking action and receiving a returned reward. During the training, because of the rarity of target events, the trained Q-network will favour the majority class more than the minority (also referred to as the data imbalance problem). A reward function is defined to control the biases during learning by assigning different rewards for various classes present. That will handle the challenge of the extreme imbalance in the dataset.

In order to train the model on the ACMS dataset, the following DRL model parameters are defined as follows.

Observation Space (S): contains all variables the agent-classifier needs to consider before classifying a data point as either positive or negative. For the problem under consideration, the agent is expected to see all the pattern variables before making a decision. The intuition here is at each given time-step, the agent-classifier is expected to consider the previous, present and future patterns before updating its weight. At the start of the training, the agent-classifier receives the first pattern as a sequence of failure/warning messages. The order of sequence is maintained so as not to alter the pattern leading to equipment maintenance. The input is in the form of a 3D array (Samples, Time Steps, and Features)

Action Space (A): The agent classifier takes action once it has assessed the environment. In our case, the action is binary $A = \{1, -1\}$ classified as positive or negative corresponding to the labels in the training dataset.

Reward (R): Represented as r_t , the reward is returned based on the action taken by the agent classifier on the environment. If the agent predicts the given pattern correctly as positive, a high reward will be returned. If it misclassifies, a penalty is given in the form of a negative value. To improve the prediction of the minority class, at each time step, a reward function is defined so that a higher reward is returned for the correct classification of the minority class and larger penalties for misclassification. This helps the agent-classifiers to become less biased towards the majority class. The reward values are chosen using the imbalance ratio defined in equation 1.

$$r_t = \begin{cases} \lambda\rho\,, & a_t = y_t \;\; where\; s_t \in D_N \\ -\lambda\rho\,, & a_t \neq y_t \;\; where\; s_t \in D_N \\ 1\,, & a_t = y_t \; where\; s_t \in D_p \\ -1, & a_t \neq y_t \;\; where\; s_t \in D_P \end{cases} \tag{7- 1}$$

where $\rho = \frac{D_P}{D_N}$, D_N is the given number of majority class elements and D_P the given number of minority class elements. λ is a trade-off parameter that allows the control of the composite to be between speed and accuracy. Where $\lambda \in [0,1]$. The range of the grid search for the parameter lambda (λ) is define in the range [0,1]. The dynamic adjustment of the reward function hyperparameters is achieved by the use of a defined function. The function is designed as part of the reward, it allows a user to specify the upper and lower values for the lambda (λ) for the model to test. The model iterate and measures the best value of lambda (λ).

Transition probability distribution dynamics (T): is the probability of transitioning from one state St to another state s_{t+1} in a single step, $p(s_{t+1}|s_t\,, a_t)$. In our case, it is deterministic the agent classifier moves from a current state s_t to the next state s_{t+1} in the sequence of patterns in the dataset.

Discount factor $(\boldsymbol{\gamma})$: The factor $\gamma \in [0,1]$, is the weight of importance of future rewards. The discount factor needs to be defined carefully since we are considering a sequence to sequence approach where a successive pattern can be related.

Exploration rate: The rate $\varepsilon = [0,1]$. It is important to explore as much of the state-action space as possible to achieve optimal policy. Therefore, we choose the e-greedy approach[51].

Episode (***e***): is the transaction trajectory of all the states that came from the initial state to the terminal state. In this solution, an episode defines as when an agent classifier reaches the end of the window.

Policy ($\boldsymbol{\pi_\theta}$): is a function that receives a sample as input and then returns the probabilities of the label, represented as the mapping function $\pi: s \rightarrow A$ where $\pi_\theta(s_t)$ denotes the action a_t performed by an agent at state s_t. In an MDP, the sequence of (s, a, r) in an episode forms a policy trajectory. End of every episode, a total cumulative reward is returned from the environment.

$$G_t = \sum_{t=0}^{T-1} \gamma^t r_{t+1} \qquad (7\text{- }2)$$

The goal of every RL algorithm is to find an optimal policy π^* Which attains the maximum expected return from all states. A policy is an agent-classifier behaviour action, and it specifies what action to take at each step. The stochastic policy is expressed as

$$\pi(a|s) = P(A_t = q, S_t = s) \qquad (7\text{- }3)$$

Where π(a│s) is the probability of taking action a in a state s under a policy π

Experience replay memory: replay memory is used in moderating the effect of the imbalance problem. The replay memory is split equally into sub-memories between classes. After the split, then each corresponding class will be appended in its memory instead of overwriting the minority sample with the overwhelming majority. This approach will ensure that when samples are randomly fetched from memory to train the agent, it will balance all the training dataset classes.

7.3.3. implementation of reinforcement learning for extremely rare failure prediction

As seen in Figure 7-3, a defined reward function (equation1) is used to provide a known reward for each action at every step. The dataset represents the environment where the agent-classifier takes an action a at a given state s (see Table 7-1), and based on the action taken, a reward is returned. The DQN addresses the fundamental instability problem of using a functional approximation in reinforcement learning (RL) by using two techniques: experience replay memory and target networks(θ). Experience replay memory stores transitions of the form Q(st, at, st+1, rt+1) in a replay buffer. This enables the agent-classifier to sample from and train on previously observed data. Not only does this massively reduce the number of interactions needed with the environment, but batches of experience can be sampled, reducing the variance of learning updates. Furthermore, the temporal correlations that can adversely affect RL algorithms are avoided by sampling uniformly from a large memory. Finally, from a practical perspective, batches of data can be efficiently processed in parallel by modern hardware, increasing throughput.

The original DQN algorithm used uniform sampling [52]. However, a later study shows that prioritizing samples based on eligibility trace [53] is more effective for learning. Q-learning seeks to find the best action to take for any finite MDP, given the current state. The Q-learning algorithm learns a policy that maximises a cumulative reward under a specific state-action pair Q(s, a). Therefore, within a given trajectory, the algorithm will perform a series of actions to obtain a maximum total reward.

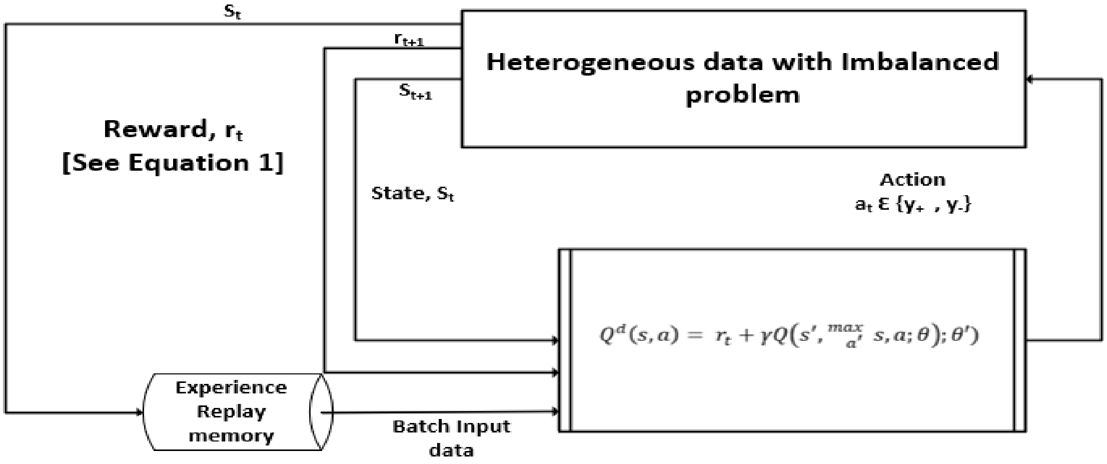

S_t
r_{t+1}
S_{t+1}
Heterogeneous data with Imbalanced problem
Reward, r_t [See Equation 1]
State, S_t
Action $a_t \in \{y_+ , y_-\}$
$Q^d(s,a) = r_t + \gamma Q(s', \overset{max}{a'} s, a; \theta); \theta')$
Experience Replay memory
Batch Input data

Figure 7- 3 Deep Reinforcement Learning for rare event prediction

I. Deep Reinforcement Learning optimal policy: An optimal policy is an integral part of the proposed DRL algorithm. Basically, in reinforcement learning, a policy is responsible for choosing an action from a given state. Therefore, an optimal policy chooses the best action from a state. Choosing the best policy is the goal of every reinforcement learning algorithm. In the proposed approach, unlike normal reinforcement learning, the agent-classifier receives an environment state as input represented by a training sample and then performs an action (classification) under the control of a policy. DRL-based classification policymaking aims to learn the classification policy that maximizes the total reward during the entire training period.

Finding an optimal policy in Q-learning, a value function is needed, and to calculate the value function a total cumulative reward (G_t) is required. To find G_t a sum of rewards for every action is needed that is

$$r_{t+1} + r_{t+2} + r_{t+3} \ldots = \sum R_t = G_t = \sum_{t=0}^{T-1} \gamma^t r_{t+1} \tag{7- 4}$$

Where T is the Trajectory.

A value function is a function that follows a policy for each step to estimate the expected future reward, expressed as

$$V_{(s)} = \mathbb{E}[G_t | S_t = s] \tag{7- 5}$$

There are two types of value functions; the state-value function (see equation 6) determines an agent's goodness in a given state. The action-value function (equation 7) determines how good it is to perform a given action in a given state.

$$V_{\pi_{(s)}} = \mathbb{E}_\pi[G_t | S_t = s] \tag{7- 6}$$

the state-value function

$$q_{(s,a)} = \mathbb{E}_\pi[G_t | S_t = s, A_{t=a}] \tag{7- 7}$$

the action-value function

An optimal policy π^* is the maximum expected reward for each state express as

$$\pi^*(a|s) = \max_\pi Q_\pi(s, a) \tag{7- 8}$$

The next step is to find a method to predict possible future rewards, but the challenge is that possible actions at future time-steps are unknown. Since the Bellman equation helps in calculating Q* at each time step, it gives a way to determine the optimal policy. Therefore, the Bellman equation[54] is used to drive the optimal policy, which incorporates the possible actions' probability at future time-steps.

$$V(s)= \mathbb{E}[R' + \gamma V(S')|S_t = S] \tag{7- 9}$$

Where $S_{t+1}\ or\ S'\ is\ the\ Next\ state$ and $S_t\ or\ S\ is\ the\ Current\ state$. Equation 7-9 is the Value Function of current state = Immediate reward + value function of the next state. The Bellman for the action-value function becomes.

$$Q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V_\pi S' \tag{7- 10}$$

More detail on Bellman's expression for state-value and action-value function is explained by David Silver [55].

Substituting equation (7-9) in (7-10) to get

$$Q_\pi(s,a) \quad = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \sum_{a' \in A} \pi(a'|s) Q_{\pi(}(s',a')) \tag{7- 11}$$

As we can infer from equation 11, the optimal policy π* is to take the best action at each given state defined by $Q(s,a)$. Therefore, the optimal Q-function becomes

$$Q^{\pi^*}(s,a) = {}^{max}_{\ a} \left\{ R_s^a + \gamma \sum_{s'} P(s'|s,a) Q^{\pi^*}(s',a') \right\} \tag{7- 12}$$

Where π^* is the optimal policy mapping sequence of states to action, Q^{π^*} is the optimal Q-function of the optimal policy.

In Q-learning, the Q-function is implemented as a table of states and actions pair, and then the values are updated iteratively as the agent accumulates knowledge. A linear approximation function for updating the weight can be sufficient if the simple environment to work with is relatively small. However, if the action space is of high dimension, the number of transactions to store gets more complex, then the use of a non-linear approximation approach such as deep neural networks becomes an option.

Deep neural network function-approximation with respect to its weights θ, is referred to as a deep Q-network. The weights θ_{is} are used to approximate the value function across the whole state-action space. The interaction data (s, a, r, s') are stored in a priority experience replay buffer (PER). The classifier agent will then randomly sample a mini-batch from the PER and perform stochastic gradient descent on the Q-network by minimizing a loss function.

$$L_i(\theta_i) = \sum_{(s,a,r,\,s')\in(PER)} (y_i - Q(s,a,s'))^2 \quad (7\text{- }13)$$

$$\text{Where } y_i = \begin{cases} r & when\ \ failure\ is\ true \\ & \quad\quad . \\ & \quad\quad . \\ & \quad\quad . \\ & \quad\quad . \\ r + \gamma \overset{max}{a'} Q(s', a'; \theta_{i-1}) & when\ \ failure\ is\ false \end{cases}$$

Differentiating the loss function (equation 13) with respect to the weights θ_i we get

$$\nabla\theta_i L_i(\theta_i) = \sum_{(s,a,r,s',)\in PER} [y_i - Q(s,a;\theta_i)\nabla\theta_i Q(s,a;\theta_i)] \quad (7\text{- }14)$$

A Q(λ)-learning [43] is used to improve the algorithm learning process. In the process, SARSA learning is combined with eligibility trace [53] and incorporated into Q-learning to give a more general method that learns efficiently using time-series data. The eligibility trace considers a temporal history of the transaction (s, a, r), since we are using function approximation instead of a Q lookup table to estimate Q-values, a trace is considered for each component of the weight θ. The update is done as follows.

$$Q_{t+1} = Q_t(s,a) + \alpha\Delta_t e_t(s,a;\theta) \quad (7\text{- }15)$$

$$\theta_i = \theta_{i-1} + \alpha\Delta e_i \quad (7\text{- }16)$$

Where $\Delta_i = y_i - Q(s,a;\theta_i)$ is the SARSA error, and $e_i = \frac{\gamma\lambda e_i + \Delta Q(s,a;\theta_i)}{\Delta\theta_i}$ is the eligibility value.

II. Deep Q -learning with Prioritized Experience Replay (PER): In the normal Q-learning or DQN, the max operator uses the same value for both action selection and action evaluation[35]. This is likely to result in the selections that lead to over-optimistic estimation. Hado et al.[35] proposed double deep reinforcement learning (DDQN). The DDQN is designed to reduce over-estimation by decomposing the max operator in the target network into action selection and action evaluation. DDQN reduces the problem of over-estimation using two value functions by randomly assigning each experience to update one of the two value functions. That there are two sets of weight $\theta \; and \; \theta'$ for every update, one set of weights is used to determine its value.

$$Q^d(s,a) = r_t + \gamma Q\left(s', \overset{max}{a'}\; s, a; \theta\right); \theta' \tag{7- 17}$$

Similarly, SARSA learning is a stochastic way of using the value of the action elected by an agent in the next step instead of using max as in Q-learning.

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma Q(s',a') - Q(s,a)] \tag{7- 18}$$

Therefore, double deep SARSA can be derived by substituting equations (15) in (17) to get

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha[\Delta_t e_t(s,a;\theta); \theta'] \tag{7- 19}$$

The use of experience replay memory to store observed transactions provides capabilities for reinforcement learning agent-classifier to remember past transaction experiences [56]. In the normal experience replay approach, the transactions are from time-to-time uniformly sampled from the buffer to update the network without considering any significance of the weight θ with respect to the policy π^*(in the DQN method, the policy is obtained implicitly by calculating a $Q_{\theta}(s,a)$ function, where the parameter θ measures the goodness of the given state-action with respect to policy). However, in a prioritized experience replay (PER) approach, the algorithm weighs the samples so that “important” ones are drawn more frequently for training[57]. The important samples are then played more frequently, which neglects the problem with strong correlations between consecutive samples. This technique improves the performance of the algorithm. Therefore, we adopted PER with double deep SARSA learning and DDQN learning as a building block for our proposed framework for predicting rare failures in the aircraft maintenance system, as seen in (algorithm 1) and (algorithm 2).

```
Algorithm 1: Double Deep SARSA- Learning

Input: Training Data $\boldsymbol{D} = \{[(\boldsymbol{x}_{1,1}, \boldsymbol{x}_{1,2} \dots \boldsymbol{x}_{1,n}), (\boldsymbol{y}_1)], \dots, [((\boldsymbol{x}_{m,1}, \boldsymbol{x}_{m,2} \dots \boldsymbol{x}_{m,n}), (\boldsymbol{y}_n))]\}$
(Episode Number k, step-size n, replay period K, Size N and exponents $\alpha, \beta$ and budget T, PER =H)
Initialize replay memory $H$ (H=$\phi, \Delta = 0, p_1 = 1$)
 Initialize action-value Function Q with random weight $\theta$, e= $0$
Initialize Environment $\varepsilon$ (observe $s_0$, and choose $A_o \sim \pi_\theta(s_0)$)
For k=1 to K do
    Training data d
    Initialize state $s_0 = x_0$
        For t=1 to T do
        Observe $(s_t, R_t, \gamma_t)$, $a_t = \pi_\theta(s_t)$
        Store transaction ($s_t$,$a_t, \gamma_t, s'_t$) in H with maximal priority $p_t = \underset{i}{max} < tp_i$
        IF $t \equiv 0 \; mod \; k \; then$
            For j=1 to k do
                Sample transaction $j \sim p(j) = \frac{p_j^\propto}{\sum_i p_i^\propto}$
                Compute importance: sampling weight $\theta_j = \frac{(N.p(j))^{-\beta}}{\underset{i}{max}\theta_i}$
                Set $y_i = \begin{cases} r_j, & label_j = True \\ r_j + \alpha[\Delta_j \, e_j \, (s, a; \theta); \theta'], & and \; label_j = False \end{cases}$
                Perform gradient descent on L($\theta$) w.r.t $\theta$:
                    $L_i(\theta_j) = \sum_{(s,a,r,\, s') \in H} (y_j - Q(s_j, a_j, s'_j;\; \theta_j); \theta_j')^2$
                Update the transaction priority $p_j \leftarrow |\Delta_j|$
                Accumulate weight change and traces.
            End For loop
```

```
                Update weights $\theta_i$
                Copy weight into the target network $Q_{Target} \leftarrow \theta$
            End IF
            Choose Action $A_t \sim \pi_\theta(s_t)$
        End For loop
        If window size = w, break
End For Loop
```

```
Algorithm 2: Double Deep Q-Network
Input: Training Data $\mathbf{D} = \{[(\boldsymbol{x_{1,1}}, \boldsymbol{x_{1,2}} \dots \boldsymbol{x_{1,n}}), (\boldsymbol{y_1})], \dots, [((\boldsymbol{x_{m,1}}, \boldsymbol{x_{m,2}} \dots \boldsymbol{x_{m,n}}), (\boldsymbol{y_n}))]\}$
(Episode Number k, step-size n, replay period K, Size N and exponents $\alpha, \beta$ $and$ budget T, PER =H)
Initialize replay memory $H$ (H=$\phi, \Delta = 0, p_1 = 1$)
 Initialize action-value Function Q with random weight $\theta$, e= 0
Initialize Environment $\varepsilon$ (observe $s_0$, $and\ choose\ A_o \sim \pi_\theta(s_0)$)
For k=1 to K do
    Training data d
    Initialize state $s_0 = x_0$
        For t=1 to T do
        Observe $(s_t, R_t, \gamma_t)$, $a_t = \pi_\theta(s_t)$
        Store transaction $(s_t, a_t, \gamma_t, s'_t)$ in H with maximal priority $p_t = \underset{i}{max} < tp_i$
        IF $t \equiv 0\ mod\ k\ then$
            For j=1 to k do
                Sample transaction $j \sim p(j) = \frac{p_j^{\propto}}{\sum_i p_i^{\propto}}$
                Compute importance: sampling weight $\theta_j = \frac{(N.p(j))^{-\beta}}{\underset{i}{max}\theta_i}$
                Set $y_i = \begin{cases} r_j, & label_j = True \\ r_j + \gamma\ \underset{a}{max} Q(s_{j+1}, a'; \theta), & and\ label_j = False \end{cases}$
                Perform gradient descent on L($\theta$) w.r.t $\theta$:
                    $L_i(\theta_j) = \sum_{(s,a,r,s')\in H} \left(y_j - Q(s_j, a_j, s'_j;\ \theta_j); \theta_j'\right)^2$
                Update the transaction priority $p_j \leftarrow |\Delta_j|$
                Accumulate weight change and traces.
            End For loop
            Update weights $\theta_i$
            Copy weight into the target network $Q_{Target} \leftarrow \theta$
        End IF
        Choose Action $A_t \sim \pi_\theta(s_t)$
    End For loop
    If window size = w, break
End For Loop
```

7.4 Experiment

An experiment is set up to investigate the application of different deep reinforcement learning (DRL) architectures for the extreme rare failure prediction problem. The transformed DRL framework's implementation is based on the proposed DDSARSA and DDQN for extremely rare failure prediction. The implementation is based on the following.

I. DQN (Baseline): This is a normal deep Q-Network that uses a neural network to approximate a state-value function in a Q-learning framework. The baseline uses a standard experience replay memory.

II. DDQN+PER: In this implementation, we use the proposed double deep Q-learning with Prioritized Experience Replay memory to predict rare event failures in aircraft predictive maintenance modelling. The aim is to investigate the effectiveness of using DDQN+PER (see algorithm 2) by evaluating the effect of the overestimation problem and efficiency of the model in handling the extreme imbalanced data.

III. DDSARSA+PER: In this implementation, we use a DDSARSA with Prioritized Experience Replay memory to predict rare event failure in aircraft predictive maintenance modelling. The aim is to investigate the effectiveness of using double deep SARSA learning with PER (see algorithm 1) by evaluating the effects of PER and eligibility trace during learning and the model's efficiency in handling the extreme imbalanced problem.

IV. To investigate the proposed deep reinforcement learning approach's performance compared to other existing rare failure prediction methods. Two methods were considered: the Cost-Sensitive method [31] and SMOTE with random forest[58]. The cost-sensitive method is an existing technique that modifies the loss function in Long Short Term Memory (LSTM) networks. The algorithm responds favourably to both classes during learning. The method is designed to handle rare failure prediction in time series datasets as implemented in previous work [31]. SMOTE+RF is a technique that balances the dataset using the Synthetic Minority Oversampling Technique (SMOTE) before presenting it as an input to the learning algorithm (Random Forests). The method is designed to handle extreme imbalanced classification problems [58]. AE-BGRU [59] is a strategy for predicting rare failure that uses a rescaled loss function in a hybrid deep network architecture known as the auto-encoder bidirectional gated recurrent network (AE-BGRU) model.

7.4.2 Description of the network architectures

There are two core approaches to data-driven maintenance, each geared towards different connected capabilities of aircraft or components. The network architecture consists of convolutional layers (CNN) and long-short team memory (LSTM) layers, which enhances the learning of the temporal dependency in the sequential data. A dense layer is also used to minimize the effect of overfitting, as seen in Table 7-2. The number of hidden layers and architecture design differ depending on the aircraft family dataset structure. For instance, an A320 aircraft will not transmit the same operational data level as the A320; hence, each dataset's learning strategies are different.

Table 7- 2 Deep Network Network Architecture

Values	Layers
	Sequential
filters =32, kernel =3, activation = ReLU	Convolution 2D
	MaxPooling
Unit =32, dropout =0.2, activation = ReLU	LSTM
Unit = 1, activation =sigmoid	Dense (Fully-connected)

Where ReLU returns X if the value is positive else, it returns zero. Max-pooling is added after the convolutional layer reduces the feature map that is generated by the convolution operation. Max-pooling also helps in selecting only important information, which removes weak activation information hence avoiding overfitting problems. LSTM layer is added to correlated information from the past with current combined with the convolutional layer helps to learn better correlations between variables. The dense layer, also referred to as fully connected, is added as the last layer it is used to make the final decision based on the input from the LSTM layer.

7.5 Results and Discussion

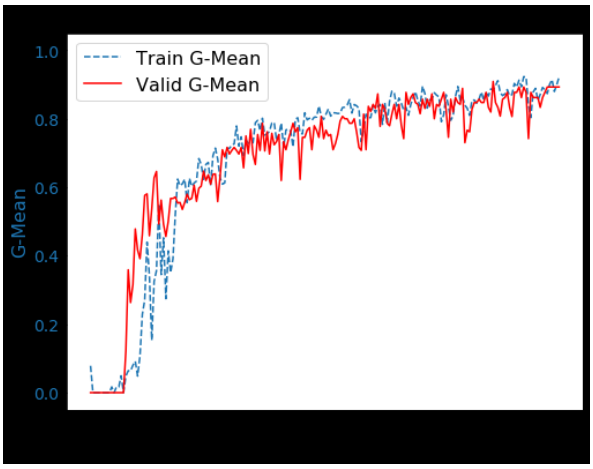
For each target event, each algorithm was run five times with the same hyperparameter for 200 epochs using five random seeds. The Q-function is approximated using deep neural networks.

7.5.1 Results

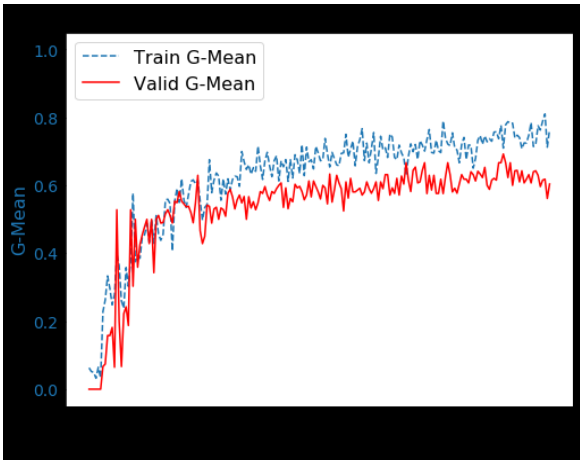
The first investigation performed was to verify the applicability and effectiveness of using DDSARSA+PER and the Deep Q-network for rare failure prediction. As seen in Table 7-3, these investigations' results are compared with a baseline method DQN (deep Q-Network). The result indicates that DDSARSA+PER and DDQN+PER can effectively be applied for rare failure prediction or data Imbalanced classification. It can generally be observed that the two novel implementations show significant improvement in terms of model performance. Although there is a delay in the training time, there is a significant reduction in both the FPR and FNR, which is very important for aircraft maintenance applications. The impact of eligibility trace positively impacts the new algorithms by reinforcing entire sequences of actions from a single experience, contributing to the improved performance in the proposed algorithms.

Table 7- 3 Shows DDSARSA learning with PER and DDQN with PER for rare failure prediction.

Aircraft ACMS Dataset											
			DQN (Baseline)			DDQN+PER			DDSARSA+PER		
	LRU	ρ	G-mean	FPR	FNR	G-mean	FPR	FNR	G-mean	FPR	FNR
A330-Family	4000KS	0.0043	0.77	0.0023	0.021	0.85	0.0015	0.004	0.94	0.00023	0.0100
	4001HA	0.0047	0.79	0.0021	0.018	0.86	0.0013	0.003	0.97	0.00023	0.0800
	5RV1	0.0044	0.78	0.0020	0.017	0.84	0.0017	0.004	0.95	0.00012	0.0111
A320 Family	11HB	0.0028	0.71	0.0025	0.023	0.82	0.0014	0.0011	0.90	0.00009	0.0000
	10HQ	0.0031	0.75	0.0021	0.019	0.84	0.0011	0.0125	0.93	0.00009	0.0000
	1TX1	0.0064	0.80	0.0020	0.018	0.89	0.0010	0.011	0.98	0.00002	0.0001

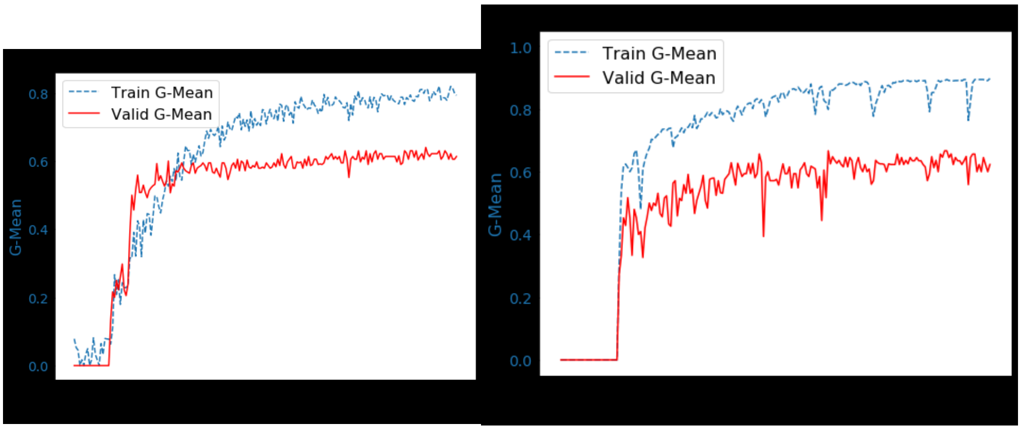


(a)



(b)

Figure 7- 4 Summary of the model performance in terms of G-Mean (data from A330 aircraft family, for component 4001HA) (a) double Deep SARSA (b) Double Deep Q_Network with prioritized experience replay memory model



(a) (b)

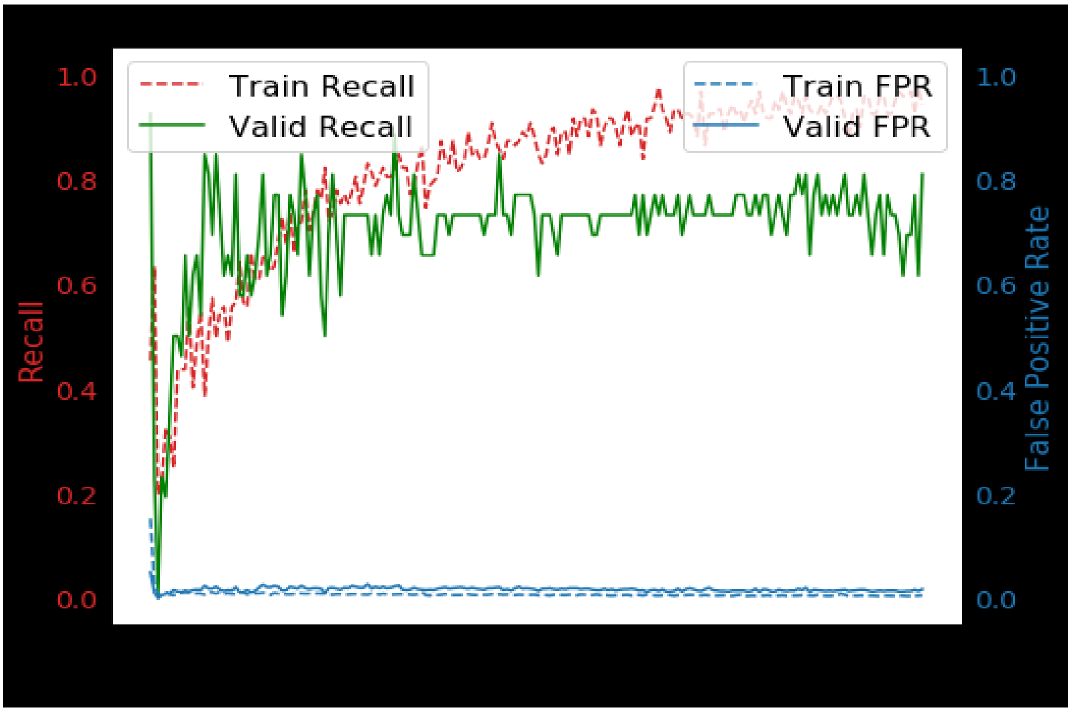
Figure 7- 5 Summary of model performance in terms of G-Mean data (A320 aircraft family for the component 1TX1) (a) double Deep SARSA (b) Double Deep Q_Network with prioritized experience replay memory model

Figures 7-4 and 7-5 show the classifier-agent performance over the validation dataset for both A330 and A320 aircraft, respectively. The model is trained for up to 200 epochs, rewarded with the parameter ρ as seen in Table 7-3, and a learning rate of 0.01. Figure 7-4(a) shows the performance of the DDSARSA+PER model, and it can be observed that the agent learns slowly between 0- 25 epochs for validation. After 25 epochs, the performance increases steadily and normalizes at 0.7g-mean for validation. Similar performance is seen in Figure 7-4(b), which shows the performance of DDQN+PER, the model learns slowly up to 15 epochs, and the performance increases steadily, achieving 0.65g-mean for validation. The model's performance on A320 aircraft is seen in Figures 7-5(a) and 7-5(b); as observed, the DDSARSA+PER model shows better G-mean performance than DDQN+PER.

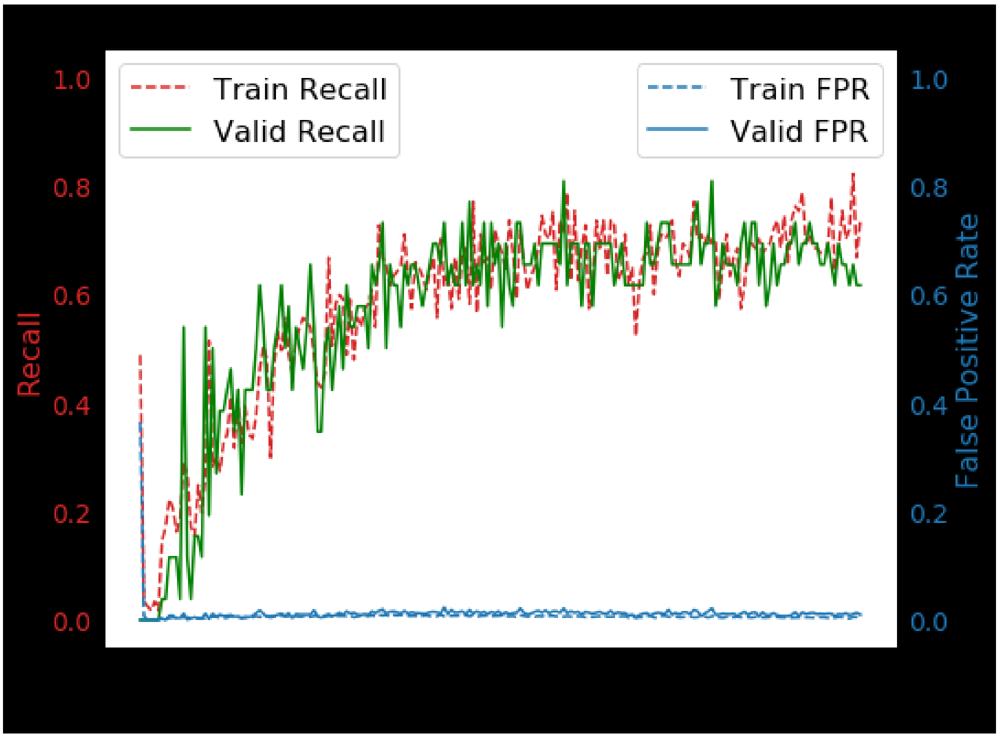
It is important to note that the choice of hyper-parameter λ and the imbalance ratio ρ significantly impact the model's overall performance because they can cause the agent to learn a sub-optimal policy. When the value of λ is large, the model converges quicker at the G-mean's expense, and when the value is small, the model converges slower with better performance. DDSARSA+PER only gives better performance at a certain value of λ based on the structure and complexity (i.e. the length of the sequence pattern for each failure) on the dataset in question. Adjusting and keeping the reward function’s parameter lambda (λ) static impacts the algorithm's performance. Therefore, to improve learning on the proposed approach, we performed a grid search in each training phase to dynamically adjust the hyper-parameters reward function (λ) based on the use-case imbalance ratio ρ. The parameter lambda (λ) is define in the range [0,1]

7.5.1.1 Model sensitivity analysis

Figures 7-6 and 7-7 illustrate model performance results based on the recall and FPR for training and validation on the A330 and A320 aircraft families. From each dataset family, one component was picked. A330's 4001HA (high-pressure bleed valve) and A320's 1TX1 (air traffic control unit).



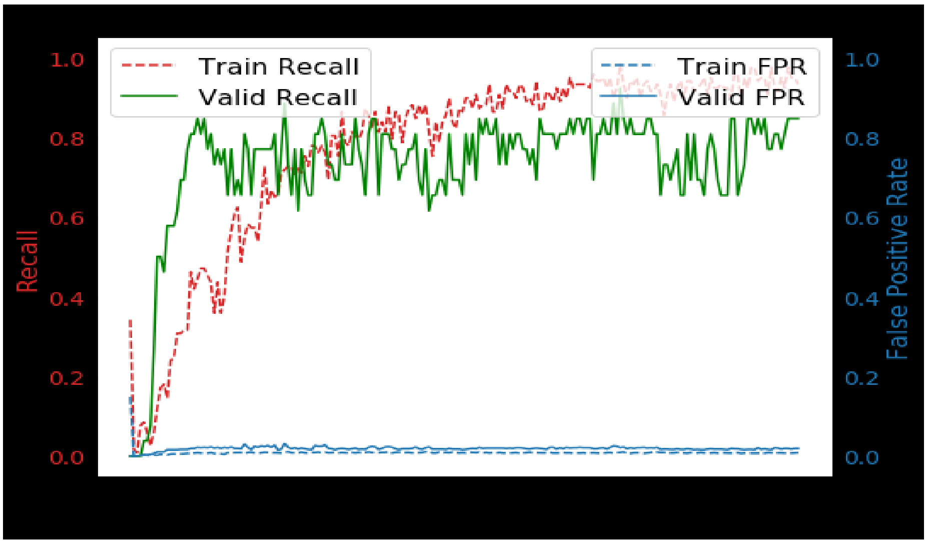
(a)



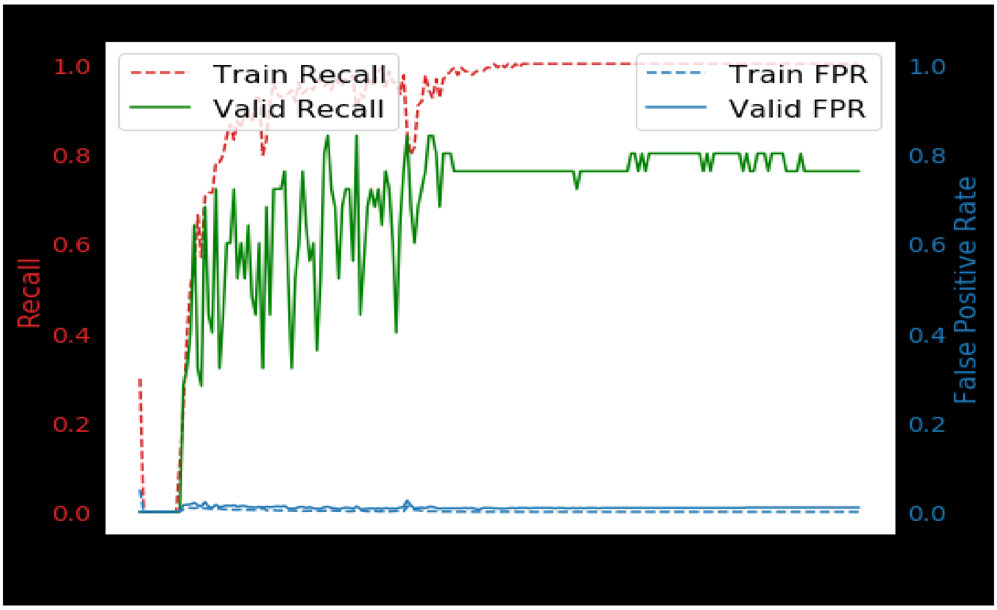
(b)

Figure 7- 6 Summary of the model performance in terms of false-positive rate (data from A330 aircraft family for the component 4001HA) (a) double Deep SARSA (b) Double Deep Q_Network with prioritized experience replay memory model.

The result of training the DDSARSA+PER algorithm using the A330 dataset is shown in Figure 7-6(a). the result indicates that it takes roughly 150 epochs to reach 0.85 recall for the validation data, with a consistent FPR of about 0.00023 during the validation period. As observed, DDSARSA shows a more robust training capability than the DDQN+PER in 7-6(b) with an FPR of 0.0013 and validation recall of 0.76.



(a)



(b)

Figure 7- 7 Summary of the model performance in terms of false-positive rate (data from A320 aircraft family for the component 1TX1) (a) double Deep SARSA (b) Double Deep Q_Network with prioritized experience replay memory model.

The model validation for both the DDSARSA+PER and DDQN+PER methods utilising the A320 aircraft family dataset is shown in Figure 7-7. Figure 7-7(a) demonstrates how the DDSARSA+PER model, with a validation score of 0.85 recall and an FPR of 0.00002, converges quicker after 100 epochs and shows a better training capability than the DDQN+PER with a validation score of 0.80

recall and an FPR of 0.011. It is worth noting that the models in both implementations (DDSARSA and DDQN) had a low false-positive rate for all test situations. However, DDSARSA+PER, on the other hand, offers the advantage of faster convergence and robustness in handling extremely imbalanced problems, as shown in g-mean and FPR scores.

Furthermore, false alarms in equipment predictive maintenance might result in increased maintenance expenditures due to unnecessary checks. It may also lower the level of trust in the equipment prognostics system. As a result, the goal is to keep FPR and FNR as low as possible while maintaining a solid G-mean. The proposed approach (DDSARSA+PER) and other existing imbalance learning methods (Cost-sensitive ensemble and random forest with SMOTE) are compared in terms of False Positive Rate (FPR), as shown in Figure 7-8.

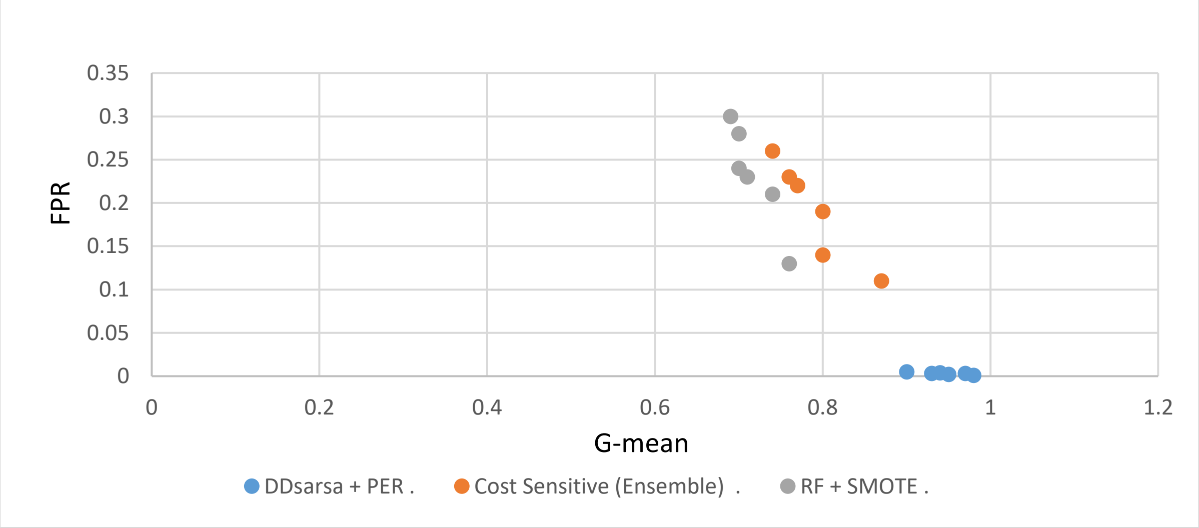


Figure 7- 8 Performance Analysis in terms of False Positive Rate (FPR) between the proposed algorithm other existing state-of-the-earth imbalance learning methods

In terms of FPR, the proposed method (DDSARSA+PER) outperformed both the algorithm level (cost-sensitive learning) [31] and the data level (Random Forest with Synthetic oversampling)[58] methods. Figure 7-8 shows that the FPR for the DDSARSA+PER is less than 0.001 in all situations studied, while cost-sensitive approaches have FPRs ranging from 0.11 to 0.26, and SMOTE+RF methods have FPRs ranging from 0.13 to 0.3. In terms of G-mean and FPR, the overall result demonstrates that the Cost-Sensitive approach and the SMOTE+RF method perform similarly on both datasets (A330 and A320 aircraft).

5.1.2 Model Validation in Predicting Failure Within a given Range

Further research was conducted to establish the model's ability to anticipate aircraft component failure within the specified time frame, such as the ability to predict a number of flights ahead of failure. It is critical to make predictions within a realistic time frame, not too far ahead of the failure point to prevent wasting resources, and not too near to the failure point to allow enough time to plan maintenance. As a result, ten to two flights prior to a failure point is considered a reasonable period for raising an alert.

Figure 7-9 depicts a graphical picture of the timeframe that leads to failure. Point zero denotes the actual failure point, whereas points less than zero (negative) denote flights prior to the failure and points larger than zero (positive) denote flights following the failure. The following requirements were considered when using the DDSARSA+PER model and ACMS testing data (representing data from previous flights without labels) to make predictions: any failure alert that arrives earlier than -10 is considered too early, and any failure alert that arrives later than -2 flights before the actual failure point (zero) is considered too late prediction.

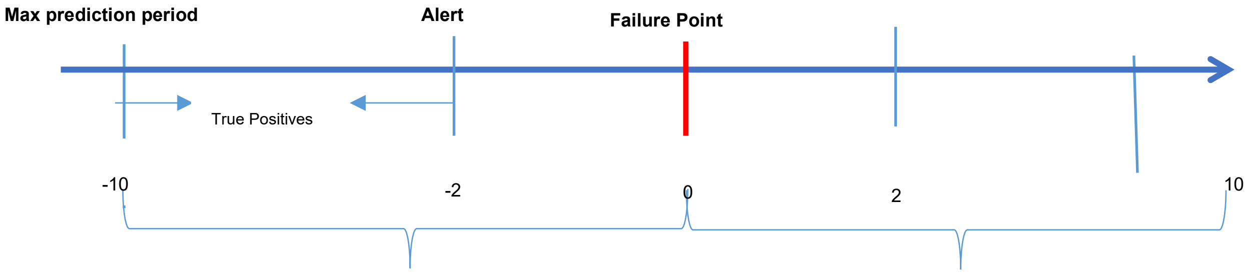


Figure 7- 9 Flight cycles before (indicated with nagive sign) and after failure

The predicted results are displayed in Figure 7-10. Each point represents the difference between the time of actual maintenance action and its predicted time (prediction residual). The residual error between two and ten (shown by the red lines) are true positives; that is, the model predicted component replacement within the desired range. Those above ten indicate the prediction came too early, and those below two indicate the model predicted maintenance too late. Residual error at point zero naturally represents the points for which maintenance and prediction were simultaneous, and negative values show a very late prediction.

It can be seen that the majority of the failure alerts for the component 4000KS (electronic engine unit) are within the target range. Only three alerts came too early, and one alert was predicted very late. For the component 4001HA (pressure regulating valve), three alerts are predicted too early, and two are predicted late, with two at precisely on the failure leg (zero), and two were predicted very late (below zero). Likewise, for the component 5RV1 (satellite data unit), the model predicted most of the failures within the target range. Similar performance is seen in the A320 aircraft family, with the model predicting a majority of failures for 11HB (the flow control valve), 10HQ (avionics equipment ventilation computer), and the 1TX1 (air traffic service unit) within the target range.

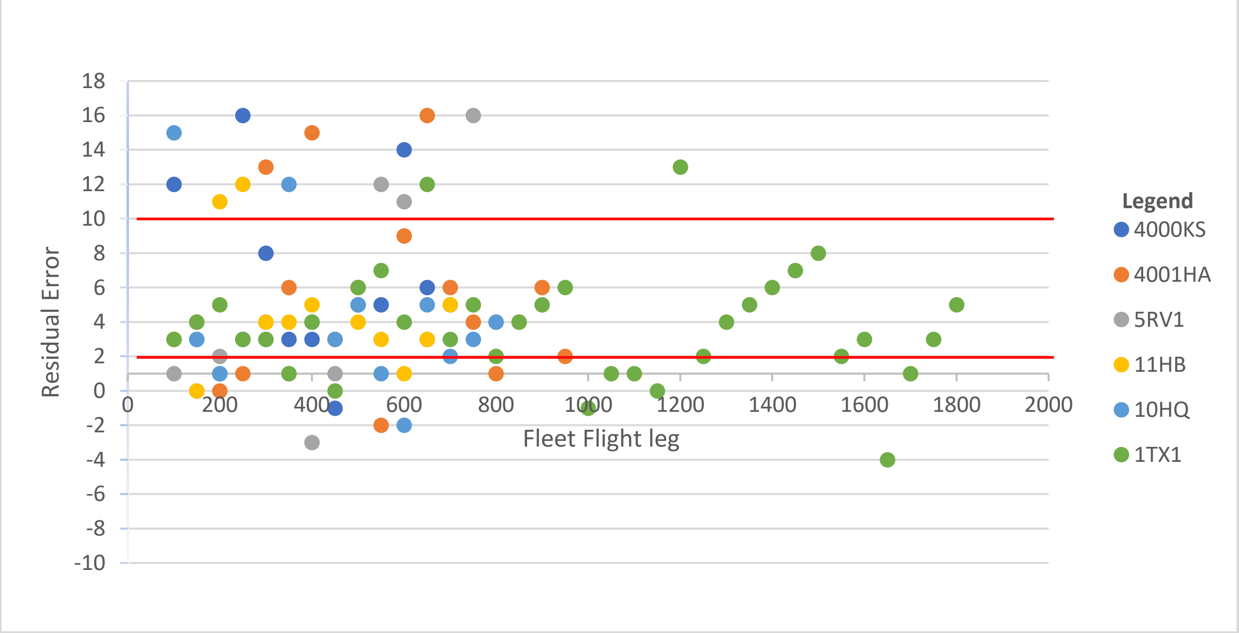
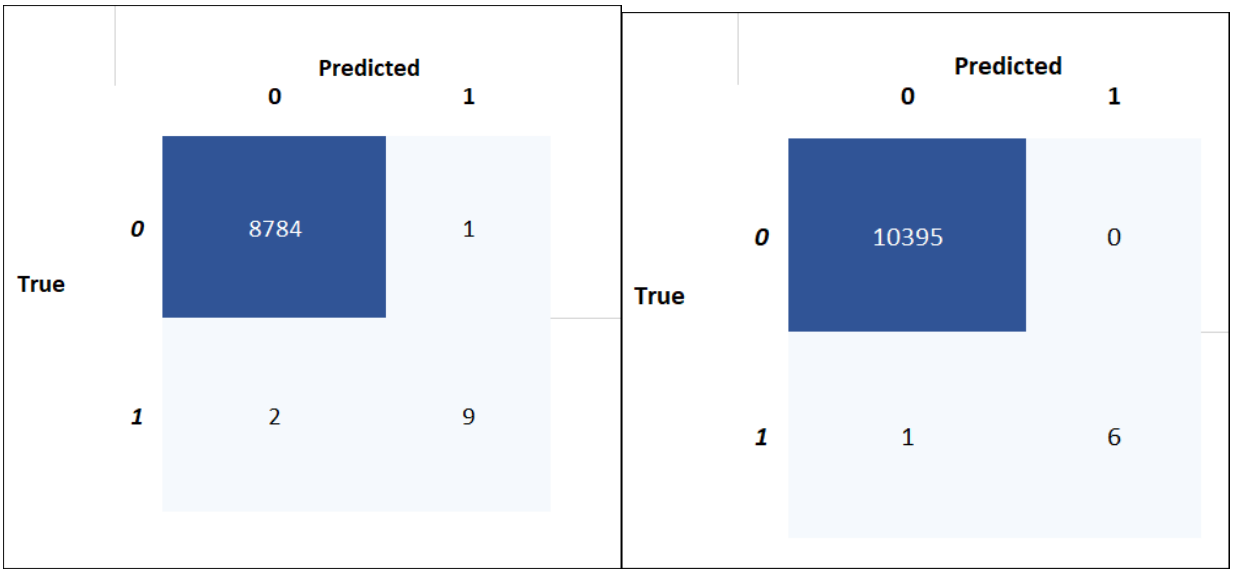


Figure 7- 10 Validation of Proposed Model against actual maintenance record

In conclusion, based on the prediction score in Figure 7-10, the proposed DDSARSA+PER model can forecast approximately 90% of aircraft component replacements within a specific range, i.e. not more than ten flights and not fewer than two flights to failure.

The number of failure cases classified is shown in Figures 7-11 (a) and 7-11 (b). The proposed model's confusion matrix was created using one component from the A330 and A320 datasets. As shown in Figure 7-11(a), the DDSARSA+PER model successfully predicted 9 out of 11 unplanned electronic engine unit failures (4000KS from the A330 dataset). Figure 7-11(b) shows the model predicted 6 out of 7 flow control valve failures (11HB from the A320 dataset).



(a) (b)

Figure 7- 11 (a) 4000KS - Electronic Control Unit/ Electronic Engine Unit (b)11HB - Flow Contol Valve

7.5.1.3 Model of Comparative Analysis

The comparative analysis between the proposed method (DDSARSA+PER) and existing imbalance learning methods (cost-sensitive and SMOTE) and previously implemented method the autoencoder with bidirectional gated recurrent unit (AE-BGRU) network [59]. It can be observed that despite the extreme imbalance ratio in all the cases considered for both the A330 and A320 datasets, the proposed method performs much better in terms of G-mean and FPR. For example, considering 4000KS with the lowest imbalance of 0.0043, it can be observed that the G-mean for DDSARSA+PER is 94% while that of cost-sensitive is 74%, SMOTE+RF is 70% and 66%. A similar performance is seen for other components, with a higher imbalance ratio compared to 4000KS. This clearly shows performance supremacy for the DDSARSA+PER model in predicting rare failure.

The performance improvement in the deep reinforcement learning implementation, especially the DDSARSA+PER model, comes from a number of different factors such as the reward function, which optimize future rewards, in contrast to a machine learning model that predicts the probability of future outcomes (classification using an ensemble method and the synthetic minority oversampling techniques with random forest). Secondly, the use of PER, which, instead of uniformly sampling transactions from replay memory, employs a prioritized approach. This also entails the replay of the important transactions more frequently and hence learns more effectively.

Table 7- 4 The performance of the proposed reinforcement learning approach with other existing rare failure prediction methods

			DDSARSA + PER			Cost-Sensitive (LSTM)			SMOTE+RF			AE-BGRU		
	Component	ρ	G-mean	FPR	FNR	G-mean	FPR	FNR	G-mean	FPR	FNR	G-Mean	FPR	FNR
A330 Family	4000KS	0.0043	0.94	0.00023	0.0100	0.74	0.026	0.103	0.70	0.024	0.022	0.66	0.0083	0.3800
	4001HA	0.0047	0.97	0.00023	0.0800	0.80	0.014	0.111	0.74	0.021	0.024	0.63	0.0013	0.4615
	5RV1	0.0044	0.95	0.00012	0.0111	0.76	0.023	0.106	0.71	0.023	0.022	0.65	0.0008	0.4000
(A320) Family	11HB	0.0028	0.90	0.00009	0.0000	0.77	0.022	0.101	0.69	0.030	0.023	0.62	0.0019	0.5555
	10HQ	0.0031	0.93	0.00009	0.0000	0.80	0.019	0.104	0.70	0.028	0.022	0.65	0.0028	0.5454
	1TX1	0.0064	0.98	0.00002	0.0001	0.87	0.011	0.101	0.76	0.013	0.017	0.81	0.0002	0.2352
Average prediction score			0.95	0.00013	0.0168	0.79	0.0192	0.1043	0.71	0.023	0.021	0.67	0.0025	0.4296

7.5.2 Discussion

The main aim of this study is to investigate the applicability of deep reinforcement learning techniques for training an extremely rare failure predictive model instead of the widely used machine learning or deep learning methods for slightly imbalanced datasets. Two algorithms (the DDSARSA and the DDQN) are designed and implemented. The implementation results show that the application of deep reinforcement learning for extremely rare failure prediction is viable, and the constructed algorithm shows superior performance as compared with baseline DQN. Also, it was observed that the proposed DDSARSA+PER algorithm shows better learning as compared to DDQN+PER. Then DDSARSA+PER was compared with existing imbalanced learning methods, and performance was evaluated based on G-mean and false-negative rates. As indicated in Table 5, the average prediction rate for all six components was calculated, in comparison to the cost-sensitive LSTM approach with 0.79 g-mean and 0.019 FPR, SMOTE+RF with 0.71 g-mean and 0.022 FPR, and autoencoder with a bidirectional gated unit network with 0.67 g-mean and 0.0026

FPR, the proposed DDSARSA showed superior performance with an overall 0.95 g-mean score and an average of 0.0005 FPR.

Calculating the percentage increase in G-mean scores from the highest g-mean, the Cost-Sensitive (LSTM) model, which is 0.79, to the highest g-mean, DDSARSA+PER, which is 0.95, the result shows that DDSARSA exhibits a 20.3% g-mean improvement. In addition, when computing the percentage drop, the suggested model reduces FPR by roughly 97.3684 % by using the lowest FPR, which is 0.019 to 0.0005.

The overall observation shows that the Cost-Sensitive method and the oversampling (SMOTE+RF) method perform relatively the same on both datasets (A330 and A320 aircraft) in terms of G-mean and FPR. What accounts for the significant performance improvement in DDSARSA are basically the combination of the convolutions in deep neural networks which enhance learning relationships between variables in the dataset, the reward function which helps to counter bias during model training and the use of prioritised experience replay memory, instead of uniformly sampling transactions from replay memory, employs a prioritised approach; this also entails replaying the important transactions more frequently, optimising the learning process. Also, DRL uses a reward function to optimise future rewards, in contrast to a machine learning (regression or classification) model that predicts the probability of future outcomes. Therefore, it can be concluded that deep reinforcement learning methods are ideally best for imbalanced classification problems because of their learning mechanism and specific learning environment and reward function. The PER and eligibility trace also contributed to the performance impact. The impact of eligibility trace positively impacts the new algorithms by reinforcing entire sequences of actions from a single experience, contributing to the improved performance in the proposed algorithms.

The impact of false alarms FNR - the proportion of "healthy" components classified as failures in equipment' predictive maintenance can result in higher maintenance costs due to unnecessary checks. Also, FPR - the proportion of faulty components classified as non-faulty or when the model fails to predict failure can result in equipment damage or huge loss. A high FPR score or FNR score might potentially lower the level of trust in the equipment prognostics system. As a result, the goal is to bring both FNR and FPR down to an acceptable level. This implies the model should accurately identify fewer false alarms, lowering total operational costs and increasing vehicle availability and reliability. As shown in Figure 7-8 in comparing the proposed model to existing approaches, the proposed DDSARASA+PER model shows a lower false-negative rate. The usage of a double deep neural network is the main disadvantage of DDSARSA+PER, which increases training time but can be compensated for by a high detection rate. This study will impact research towards mitigating unscheduled maintenance for systematic schedule maintenance.

7.6. Conclusion

In this study, a novel technique for predicting extremely rare failure is proposed and implemented. The new technique is based on a deep reinforcement learning approach. Two algorithms are constructed, the double deep Q-Network with prioritized experience replay memory and the double deep state-action-reward-state-action with prioritized experience replay memory. The effectiveness of the new approach is validated using a real-world aircraft central maintenance log-based dataset. The result shows that the application of deep reinforcement learning for extremely rare failure prediction is viable. It also indicates that the proposed double deep state-action-reward-state-action with prioritized experience replay memory model can effectively predict component failure in both the A330 and A320 aircraft families with low false-positive and false-negative rates. The result means that unscheduled maintenance can be reduced in the aircraft fleet at the same time decreasing the cost of maintenance operations.

The work can be extended by carrying out further experimentation to determine the impact of high imbalanced on other deep reinforcement learning. Parameters such as changing the network architecture, an additional variable can be introduced into the deep neural network to keep track of the physical state and check for inconsistency with the physical laws to improve accuracy. Also, future work can consider enhancing performance optimization using other deep reinforcement learning algorithms. An ablation study will be carried out to assess the impact of eligibility trace and prioritise experience replay memory individually. More aircraft data sources - such as quick access recorder (QAR) Data, Performance Reports (PR), and Maintenance Tech Logs data can be integrated into the analysis.

7.7 Acknowledgement

The authors would like to acknowledge the Integrated Vehicle Health Management Centre (IVHM), Cranfield University, and the first author would like to thank PTDF Nigeria for sponsoring his study.

7.8 Reference

1. Korvesis P. Machine Learning for Predictive Maintenance in Aviation. 2017. Available at: DOI:theses.fr/2017SACLX093

2. Martinez C., Perrin G., Ramasso E., Rombaut M. A deep reinforcement learning approach for early classification of time series. European Signal Processing Conference. EURASIP; 2018; 2018-Septe: 2030–2034. Available at: DOI:10.23919/EUSIPCO.2018.8553544

3. Lin E., Chen Q., Qi X. Deep reinforcement learning for imbalanced classification. Applied Intelligence. 2020; Available at: DOI:10.1007/s10489-020-01637-z

4. Leevy JL., Khoshgoftaar TM., Bauder RA., Seliya N. A survey on addressing high-class imbalance in big data. Journal of Big Data. Springer International Publishing; 2018; 5(1). Available at: DOI:10.1186/s40537-018-0151-6

5. Ran Y., Zhou X., Lin P., Wen Y., Deng R. A Survey of Predictive Maintenance: Systems, Purposes and Approaches. 2019; XX(Xx): 1–36. Available at: http://arxiv.org/abs/1912.07383

6. Patel H., Singh Rajput D., Thippa Reddy G., Iwendi C., Kashif Bashir A., Jo O. A review on classification of imbalanced data for wireless sensor networks. International Journal of Distributed Sensor Networks. 2020; 16(4). Available at: DOI:10.1177/1550147720916404

7. Burnaev E. Rare Failure Prediction via Event Matching for Aerospace Applications. 2019 3rd International Conference on Circuits, System and Simulation, ICCSS 2019. 2019; : 214–220. Available at: DOI:10.1109/CIRSYSSIM.2019.8935598

8. François-lavet V., Henderson P., Islam R., Bellemare MG., François-lavet V., Pineau J., et al. An Introduction to Deep Reinforcement Learning. (arXiv:1811.12560v1 [cs.LG]) http://arxiv.org/abs/1811.12560. Foundations and trends in machine learning. 2018; II(3–4): 1–140. Available at: DOI:10.1561/2200000071.Vincent

9. Çinar ZM., Nuhu AA., Zeeshan Q., Korhan O., Asmael M., Safaei B. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. Sustainability (Switzerland). 2020; 12(19). Available at: DOI:10.3390/su12198211

10. Daigle MJ., Goebel K. Model-based prognostics with concurrent damage progression

processes. IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans. 2013; 43(3): 535–546. Available at: DOI:10.1109/TSMCA.2012.2207109

11. Wu D., Jennings C., Terpenny J., Gao RX., Kumara S. A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests. Journal of Manufacturing Science and Engineering. 2017; 139(7): 071018. Available at: DOI:10.1115/1.4036350

12. Schwabacher M. A Survey of Data-Driven Prognostics. Infotech@Aerospace. 2005; (May). Available at: DOI:10.2514/6.2005-7002

13. Woodward PW., Castillo E. Extreme Value Theory in Engineering. The Statistician. 1993. 79 p. Available at: DOI:10.2307/2348127

14. Falk M. Multivariate Extreme Value Theory and D-Norms. 2019. 241 p. Available at: DOI:10.1007/978-3-030-03819-9

15. Kamarujjaman., Maitra M., Chakraborty S. A novel decision-based adaptive feedback median filter for high density impulse noise suppression. Multimedia Tools and Applications. Multimedia Tools and Applications; 2020; Available at: DOI:10.1007/s11042-020-09473-6

16. Rydman M. Application of the Peaks-Over-Threshold Method on Insurance Data. Uppsala Universitet U.U.D.M. Project Report. 2018; 32: 1–21.

17. Murphy KP. Machine Learning A Probabilistic Perspective. The MIT Press. 2012. Available at: DOI:10.1007/978-94-011-3532-0_2

18. Bzdok D., Altman N., Krzywinski M. Points of Significance: Statistics versus machine learning. Nature Methods. Nature Publishing Group; 2018; 15(4): 233–234. Available at: DOI:10.1038/nmeth.4642

19. Laptev N., Yosinski J., Erran Li L., Smyl S., Li EL., Smyl S. Time-series Extreme Event Forecasting with Neural Networks at Uber. International Conference on Machine Learning - Time Series Workshop. 2017; (34): 1–5. Available at: http://roseyu.com/time-series-workshop/submissions/TSW2017_paper_3.pdf

20. Allen RJ., Valeriani C., Rein Ten Wolde P. Forward flux sampling for rare event simulations. Journal of Physics Condensed Matter. 2009; 21(46). Available at: DOI:10.1088/0953-

8984/21/46/463102

21. Berberidis C., Angelis L., Vlahavas I. Inter-transaction association rules mining for rare events prediction. Proc. 3rd Hellenic Conference …. 2004; Available at: http://lpis.csd.auth.gr/publications/076-Berberidis-Angelis-Vlahavas-SETN04.pdf

22. Sammouri W., Côme E., Oukhellou L., Aknin P., Fonlladosa C-E. Floating train data systems for preventive maintenance: A data mining approach. Proceedings of 2013 International Conference on Industrial Engineering and Systems Management, IEEE - IESM 2013. 2013.

23. Arulkumaran K., Deisenroth MP., Brundage M., Bharath AA. Deep reinforcement learning: A brief survey. IEEE Signal Processing Magazine. 2017; 34(6): 26–38. Available at: DOI:10.1109/MSP.2017.2743240

24. Moniz N., Monteiro H. No Free Lunch in imbalanced learning. Knowledge-Based Systems. Elsevier B.V.; 2021; 227: 107222. Available at: DOI:10.1016/j.knosys.2021.107222

25. Fernández Alberto, Garcia Salvador, Galar Mikel, Prati Ronaldo, Krawczyk Bartosz HF. Learning From Imbalanced Data Sets. 2018. Available at: DOI:https://link.springer.com/content/pdf/10.1007%2F978-3-319-98074-4.pdf (Accessed: 6 May 2019)

26. Chawla N V., Bowyer KW., Hall LO., Kegelmeyer WP. SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research. 2002; 16: 321–357. Available at: DOI:10.1613/jair.953

27. Lusa L., Others. SMOTE for high-dimensional class-imbalanced data. BMC bioinformatics. 2013; 14(1): 106. Available at: DOI:10.1186/1471-2105-14-106

28. Hu Y., Guo D., Fan Z., Dong C., Huang Q., Xie S., et al. An Improved Algorithm for Imbalanced Data and Small Sample Size Classification. J. Data Anal. Inf. Process. 2015; 03(03): 27–33. Available at: DOI:10.4236/jdaip.2015.33004

29. Haixiang G., Yijing L., Shang J., Mingyun G., Yuanyue H., Bing G. Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications. 2017; 73: 220–239. Available at: DOI:10.1016/j.eswa.2016.12.035

30. Wu Z., Lin W., Ji Y. An Integrated Ensemble Learning Model for Imbalanced Fault

Diagnostics and Prognostics. IEEE Access. 2018; 6: 8394–8402. Available at: DOI:10.1109/ACCESS.2018.2807121

31. David Dangut M., Skaf Z., Jennions I. Rescaled-LSTM for Predicting Aircraft Component Replacement Under Imbalanced Dataset Constraint. 2020 Advances in Science and Engineering Technology International Conferences (ASET). IEEE; 2020. pp. 1–9. Available at: DOI:10.1109/ASET48392.2020.9118253

32. Qi D., Majda AJ. Using machine learning to predict extreme events in complex systems. Proceedings of the National Academy of Sciences of the United States of America. 2020; 117(1): 52–59. Available at: DOI:10.1073/pnas.1917285117

33. Johnson JM., Khoshgoftaar TM. Survey on deep learning with class imbalance. Journal of Big Data. Springer International Publishing; 2019; 6(1). Available at: DOI:10.1186/s40537-019-0192-5

34. Keneshloo Y., Shi T., Ramakrishnan N., Reddy CK. Deep Reinforcement Learning for Sequence-to-Sequence Models. IEEE Transactions on Neural Networks and Learning Systems. 2019; : 1–21. Available at: DOI:10.1109/tnnls.2019.2929141

35. Van Hasselt H., Guez A., Silver D. Deep reinforcement learning with double Q-Learning. 30th AAAI Conference on Artificial Intelligence, AAAI 2016. 2016; : 2094–2100.

36. Luong NC., Hoang DT., Gong S., Niyato D., Wang P., Liang YC., et al. Applications of Deep Reinforcement Learning in Communications and Networking: A Survey. IEEE Communications Surveys and Tutorials. IEEE; 2019; 21(4): 3133–3174. Available at: DOI:10.1109/COMST.2019.2916583

37. Li C., Qiu M., Li C. Reinforcement Learning for Cybersecurity. Reinforcement Learning for Cyber-Physical Systems. 2019; (MI): 155–168. Available at: DOI:10.1201/9781351006620-7

38. Mosavi A., Ghamisi P., Faghan Y., Duan P. Comprehensive Review of Deep Reinforcement Learning Methods and Applications in Economics. 2020; (March): 1–43. Available at: DOI:10.20944/preprints202003.0309.v1

39. Chakraborty S. Capturing Financial markets to apply Deep Reinforcement Learning. 2019; : 1–17. Available at: http://arxiv.org/abs/1907.04373

40. Gijsbrechts J., Boute RN., Van Mieghem JA., Zhang D. Can Deep Reinforcement Learning Improve Inventory Management? Performance and Implementation of Dual Sourcing-Mode Problems. SSRN Electronic Journal. 2019; : 1–26. Available at: DOI:10.2139/ssrn.3302881

41. Jonsson A. Deep Reinforcement Learning in Medicine. Kidney Diseases. 2019; 5(1): 18–22. Available at: DOI:10.1159/000492670

42. Waschneck B., Reichstaller A., Belzner L., Altenmüller T., Bauernhansl T., Knapp A., et al. Optimization of global production scheduling with deep reinforcement learning. Procedia CIRP. 2018; 72: 1264–1269. Available at: DOI:10.1016/j.procir.2018.03.212

43. Lee XY., Balu A., Stoecklein D., Ganapathysubramanian B., Sarkar S. A case study of deep reinforcement learning for engineering design: Application to microfluidic devices for flow sculpting. Journal of Mechanical Design, Transactions of the ASME. 2019; 141(11): 1–10. Available at: DOI:10.1115/1.4044397

44. Knowles M., Baglee D., Wermter S. Reinforcement learning for scheduling of maintenance. Res. and Dev. in Intelligent Syst. XXVII: Incorporating Applications and Innovations in Intel. Sys. XVIII - AI 2010, 30th SGAI Int. Conf. on Innovative Techniques and Applications of Artificial Intel. 2011; : 409–422. Available at: DOI:10.1007/978-0-85729-130-1-31

45. Rocchetta R., Bellani L., Compare M., Zio E., Patelli E. A reinforcement learning framework for optimal operation and maintenance of power grids. Applied Energy. Elsevier; 2019; 241(December 2018): 291–301. Available at: DOI:10.1016/j.apenergy.2019.03.027

46. Zhang C., Gupta C., Farahat A., Ristovski K., Ghosh D. Equipment health indicator learning using deep reinforcement learning. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2019; 11053 LNAI: 488–504. Available at: DOI:10.1007/978-3-030-10997-4_30

47. Wiering M., Schmidhuber J. Fast Online Q(λ). Machine Learning. 1998; 33(1): 105–115. Available at: DOI:10.1023/A:1007562800292

48. Martin M. Bellman equations and optimal policies. Learning. 2011; Available at: DOI:https://www.cs.upc.edu/~mmartin/Ag4-4x.pdf

49. Montague PR. Reinforcement Learning: An Introduction, by Sutton, R.S. and Barto, A.G.

Trends in Cognitive Sciences. 1999; 3(9): 360. Available at: DOI:10.1016/s1364-6613(99)01331-5

50. Rummery GA., Niranjan M. ON-LINE Q-LEARNING USING CONNECTINIST SYSTEMS. Cambridge, England: University of Cambridge, Department of Engineering. 1994; 37(9): 20. Available at: http://mi.eng.cam.ac.uk/reports/svr-ftp/auto-pdf/rummery_tr166.pdf

51. Bouneffouf D., Bouzeghoub A., Gançarski AL. Following the user's interests in mobile context-aware recommender systems: The hybrid-e-greedy algorithm. Proceedings - 26th IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2012. IEEE; 2012; : 657–662. Available at: DOI:10.1109/WAINA.2012.200

52. Mnih V., Kavukcuoglu K., Silver D., Rusu AA., Veness J., Bellemare MG., et al. Human-level control through deep reinforcement learning. Nature. Nature Publishing Group; 2015; 518(7540): 529–533. Available at: DOI:10.1038/nature14236

53. Mousavi SS., Schukat M., Mannion P. Applying Q(λ)-learning in Deep Reinforcement Learning to Play Atari Games. Ala. 2017; : 1–6.

54. Barron EN., Ishii H. The Bellman equation for minimizing the maximum cost. Nonlinear Analysis. 1989; 13(9): 1067–1090. Available at: DOI:10.1016/0362-546X(89)90096-5

55. Silver D. Markov decision processes. Advances in Computer Vision and Pattern Recognition. 2015; 54: 199–216. Available at: DOI:10.1007/978-1-4471-6699-3_11

56. Schaul T., Quan J., Antonoglou I., Silver D. Prioritized experience replay. 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings. 2016; : 1–21.

57. Liu R., Zou J. The Effects of Memory Replay in Reinforcement Learning. 2018 56th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2018. 2019; : 478–485. Available at: DOI:10.1109/ALLERTON.2018.8636075

58. Dangut MD., Skaf Z., Jennions IK. An integrated machine learning model for aircraft components rare failure prognostics with log-based dataset. ISA Transactions. Elsevier Ltd; 2021; 113: 127–139. Available at: DOI:10.1016/j.isatra.2020.05.001

59. Dangut MD., Skaf Z., Jennions IK. Rare Failure Prediction Using an Integrated Auto-encoder and Bidirectional Gated Recurrent Unit Network. IFAC-PapersOnLine. Elsevier BV; 1 January 2020; 53(3): 276–282. Available at: DOI:10.1016/j.ifacol.2020.11.045

60. Powers DMW. Evaluation : From Precision , Recall and F-Factor to ROC , Informedness , Markedness & Correlation. International Journal of Machine Learning Technology 2:1 (2011), pp.37-63. 2007; (December). Available at: DOI:arXiv:2010.16061 [cs.LG]

61. Roc B. Comparing Two ROC Curves – Independent Groups Design. NCSS, LLC. 2021; : 1–26.

CHAPTER 8: General Discussion

This chapter presents a general discussion about the methods implemented in this thesis. The methods focus majorly on detecting and predicting rare component failure in the aircraft maintenance system. Predicting rare failure in a complex system such as aircraft is challenging due to the nature of the process and the complex structure of aircraft datasets. Also, finding abnormal patterns in large log-based data is highly challenging due to the complex nonlinear relationships among the components, processes, and subsystems. The difficulty in predicting rare events in complex systems is that the event has varying intrinsic data characteristics such as distributions and irregular failure patterns, class-overlapping, small-class disjunct, which can impact the type of algorithm used to train the model. Predicting component failure is a critical problem in the aerospace industry because of its impact on business. As reported by airlines, a single unexpected component failure has a high negative impact on the business as compared to planned or scheduled replacement. Therefore, even a small percentage in reducing unplanned failure can significantly benefit the airlines. In this study, the minority class represents the unplanned component failures, which are extremely rare in the overall ACMS dataset. Hence, four different approaches for predicting extremely rare failure, handling class overlapping, and small class disjunct are proposed and implemented.

In order to develop an effective predictive maintenance model for complex systems such as aircraft, accurate data and robust machine learning methods are necessary. For instance, in developing a predictive model for aircraft component failures using ACMS data, where the intrinsic properties of each failure vary, the integration of multiple algorithms in the prediction system is essential to assist in handling the various characteristics of components failures. Figure 8-1 shows how the proposed algorithms implemented in this thesis can be used to improve failure prediction in the predictive maintenance system. First, step 1 involves feeding the prediction system with ACMS data (log-based) obtained from the same family of aircraft in the fleet. The input data will then be filtered to obtain only relevant patterns related to the target components. Then using the filtered data as input at step 2, all the algorithms will be trained and assessed to find the best model for that component. The algorithm with the best score will be accepted at step 3. Then at step 4, the resulting model can then be used to forecast the occurrences of similar failures. For example, the BACHE algorithm is good at managing severely imbalaced with class overlapping problems when it comes to predicting flow control valve valve failure. In this scenario, the BACHE algorithm will be ideal for predicting such failures.After training all the algorithms using the patterns and assessed for the best score. The best algorithm will heuristically marked for forecasting such failures in the future. The process will be the same for all target failure instances. Four distinct data-driven algorithms for rare failure detection are proposed and implemented in this study. The solutions focus on tackling data challenges such as pre-processing, extremely imbalanced classification challenges, class-overlapping, and small class disjunct problems.

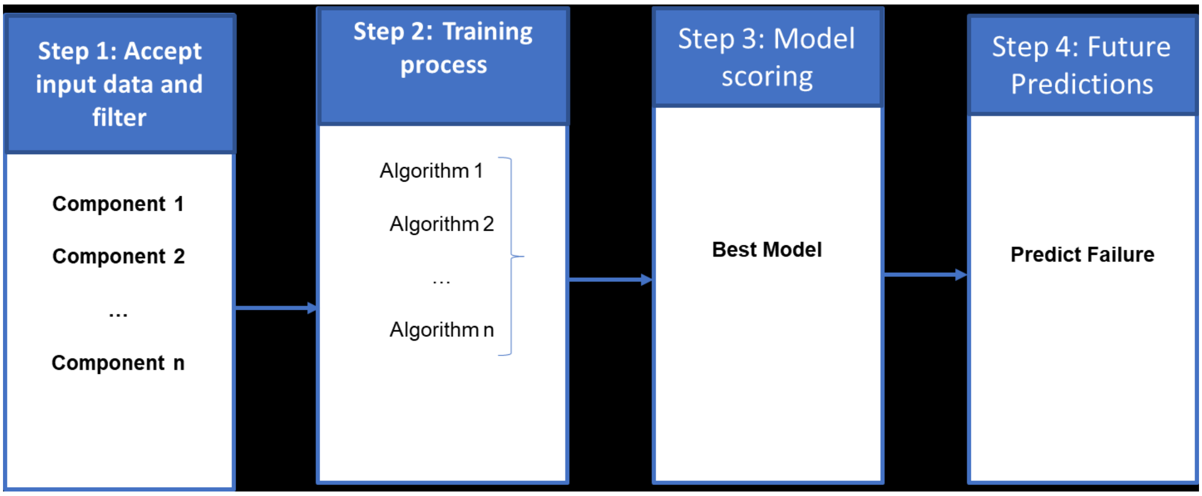


Figure 8- 1 The rare event prediction process

Furthermore, the proposed solution can give a generic solution that can be used in a variety of application domains when dealing with an imbalanced dataset, particularly when the data contains high imbalance ratios, class overlapping, and small class disjunct difficulties. The findings of the study can be applied to the creation of robust and high-performance predictive maintenance systems in a variety of industries.

The four algorithms are developed to respond to data abnormalities discussed in Chapter 2, and research gaps detailed in Chapter 3. The proposed implementations are summarised as follows:

8.1. A hybrid algorithm for pattern extraction

Patterns represent large groups of log messages which are important in performing analysis, such as event prediction or anomaly detection. Extracting patterns in large log messages from heterogeneous sources without any prior information is quite challenging. As a result, an algorithm is proposed for extracting patterns from ACMS data for rare event prediction in the predictive maintenance model. The approach combines time frequency-inverse document frequency (TF-IDF) and word vectorisation techniques used in natural language processing. A random forest technique is used to train the patterns. The main takeaway from this implementation is that the proposed methodology outperforms other similar methods in terms of pattern categorization accuracy for all the target components considered. The proposed method was compared with the closes imbalanced technique known as the Synthetic Minority Oversampling Technique -SMOTE (Figure 4-2). The new approach shows increased precision, recall, and F1-score performance. The overall score indicates that the new approach can averagely reduce 10% false positives and false negatives rates (see chapter 4).

What accounted for the performance improvement is the uniqueness of extracting the patterns and the oversampling techniques. In the new approach, related patterns are first placed closer together based on the corpus of relationships, which help in filtering out unrelated ones, making the algorithm learn better. Also, during model training, the proposed algorithm checks for patterns that regularly occur together, resulting in the component replacement (positively labelled). Then it randomly creates a possible combination of all the positively labelled patterns related to the individual target component, producing more new patterns for the minority class. This is done to increase the instance of the minority class patterns related to each replacement, which balances the training dataset to improve the detection of the rare minority samples. The combination and creation of new patterns are achieved through bootstrapping approach [1]. In contrast, the SMOTE algorithm randomly creates synthetic points by duplicating examples from the minority class to balance the class distribution before fitting a model. SMOTE usually creates synthetic examples without considering the borderline examples from the majority class, which can create an ambiguous example resulting in the class overlapping problem, leading to the model performance reduction.

The proposed algorithm's impact on the predictive maintenance model is as follows: Unplanned component failure is a major issue in the aerospace industry because of its financial implications. When compared to scheduled maintenance, the expenses of unscheduled maintenance are typically higher. As a result, every company strives to reduce these unplanned costs as much as feasible. Reduced unscheduled maintenance by even a small proportion can impact industries and businesses like airlines and MROs. Therefore, having a learning system that can correctly forecast infrequent failure can improve the prediction of aircraft component failure. As shown in Table 4.1 in Chapter 4, the results show the average prediction for various components (each FIN has different failure patterns, imbalanced ratios, and distribution). The True Positive Rate was estimated to be around 75%. (i.e. the percentage of positive examples that are correctly classified is 75%). In addition, approximately an overall 25% false-negative rate was recorded (which means the 25% of positive examples were misclassified). Although the results demonstrate an improvement in recognising rare failure, there is a significant rate of false-negative and false-positive, which is identified as the drawback of the proposed approach. Misclassifying a component's non-failure as a failure (false-negative) can result in higher maintenance costs, which is undesirable but less harmful than misclassifying a fault as a non-fault (false positive), which can result in equipment damage. As a result, more optimised methods are still necessary for such failures.

8.2. The Balanced Calibrated Hybrid Ensemble Technique

One of the fundamental research questions that this implementation seeks to answer is can a class overlapping and small disjunct problem inherent in the ACMS dataset be overcome by using hybrid ensemble learning? A new Balanced Calibrated Hybrid Ensemble Technique (BACHE) algorithm was presented to answer the above-mentioned question. The goal of the algorithm's development is to investigate how effective an ensemble-based method is in handling class overlapping and class disjunct problems in severely imbalanced datasets [2]. Ensemble learning is a methodology where multiple machine learning models are trained to solve the same problem, and the output of the learner is combined to get improved performance. The method tries to improve the machine learning classifier's performance by combining the decision of other classifiers, known as weak learners [3][4][2]. The proposed strategy tackle class overlapping concerns by splitting the data into subsets and treating each subset individually before merging the results. It employs a cascade balanced technique to decompose a class imbalance problem into a set of subproblems, each with a lower and manageable imbalance. Then, for each weak learner in each subset, a calibrated boosting with a cost-sensitive decision tree is utilised to recognise hard-to-learner patterns while avoiding the class overlapping and small class disjunct problem, improving the prediction of the extreme minority class. The approach's novelty is in the design's uniqueness as well as the fusing of the weak ensemble classifier. A cost-sensitive is utilised in each weak classifier to improve the prediction of minority class samples. The proposed approach is compared against existing ensemble learning algorithms (Balance Bagging) and hybrid imbalance learning algorithms (SMOTE + Random Forest). The baseline technique was chosen because of its design similarity to the proposed method. The proposed BACHE employs a heterogeneous cost-sensitive decision tree as a weak classifier, followed by an ensemble technique to create a hybrid algorithm (BACHE) as presented in Chapter 5. Balance bagging, on the other hand, prioritises oversampling each subset of data before fitting it with a weaker classifier [5]. In partitioning the training dataset, both approaches use boosting methods.

The main observation from the proposed approach is that the new method shows superior perforce in terms of Precision, Recall and the G-mean as compared to baseline balance bagging (see section 5.4.4). The new BACHE algorithm outperforms the baseline Balance-bagging approach for a variety of reasons, including the use of a calibrated cost-sensitive decision tree in BACHE' weak learners rather than oversampling each subset as in balanced bagging. The inclusion of cost-sensitive in the weaker learners, which is applied to all subsets, reduced the imbalance ratio, assisting in overcoming the difficulty of class overlaps. Furthermore, because the data was divided into subsets using the bootstrapping method, a subset could contain zero samples of the minority class, making the weak learners not to perform well by themselves either because they have a high bias or high variance. Hence, using an ensemble of classifiers helps tackle the challenge of the bias-variance tradeoff present in a single classifier. Also, the use of homogeneous boosting allows the BACHE algorithm to learn multiple input data distributions while simultaneously controlling high bias and variance (bias-variance trade-off), thereby becoming more robust. As a result, the ensemble classifier's performance is improved (Figure 5.3). BACHE's impact on predictive maintenance is that it has an overall false-negative rate of around 18% (Figure 5.5), compared to the prior implementation [6], which had an FNR of 20%, suggesting an 8% reduction in false negatives. The decrease in both false-negative and false-positive can decrease unscheduled maintenance, which leads to a decrease in overall system failure.

8.3. Mixed Gaussian Process with Expected Maximisation Algorithm

A study was carried out in the search for an optimisation technique for tackling the class overlapping and class disjunct problem, with a focus on aircraft predictive maintenance modelling utilising ACMS data. A cluster-based resampling method was suggested based on the Mix Gaussian Process with Expected Maximisation (MGP-EM).

Maximum likelihood estimation (MLE) estimates the parameters of an assumed probability distribution for a given dataset by searching across probability distributions and their parameters[7]. This is achieved by maximising a likelihood function to make the observed data most probable [19].

Maximum likelihood becomes useful if there exist variables that interact with those present in the dataset but were hidden or not observed, known as latent variables [19]. An expected maximisation algorithm is an algorithm that is capable of performing maximum likelihood estimation in the presence of latent variables. A mixture model is a model that is made up of a combination of many probability distribution functions, while a Mixed Gaussian process model is a mixture model that uses a variety of many normal distributions and needs estimation for mean and variance for each. The motivation for this study is because a different process generates the variables in the ACMS dataset, the examples belonging to each process have a normal probability distribution, but the combined data is overlapped as seen in Figure 2-12 (that is, the distributions for the joined data are similar enough that it is not obvious to which distribution a given an example may belong) making it difficult for the machine learning classifier.

The influence of MGP-EM on a severely imbalanced dataset with a class-overlapping problem is investigated in this work. The proposed approach identifies and groups the data according to their similarity to avoid creating small disjuncts in the learned hypothesis. The rationale behind implementing the MGPEM-based strategy is to, in the process of learning, compute explicitly the probability of points belonging to each cluster, which deals with an in-between point and avoids ambiguity problems in clustering. The proposed method is designed to overcome the problem of class-overlapping and small disjunct in the concept-learning, which is difficult for the classifier to learn, hence improving the prediction of a minority class.

In order to understand the effectiveness of the proposed approach in handling the class overlapping, the algorithm was trained using data that contains the ACMS data and the performance of the model measured in terms of Precision, Recall, F1-score, and ROC curve. The model shows an average performance of 90% precision and 80% recall, meaning the classifier's 90% true positive predictions (component replacement) are truly correct. The 89% AUC means the model has an 89% chance to distinguish between positive and negative, showing effeteness in handling class overlapping problems in an extremely imbalanced dataset.

8.4. Autoencoder Convolutional Neural Network -Bidirectional Gated Recurrent Unit Approach.

As pointed out in Chapter 3, deep learning is a branch of machine learning that consists of numerous processing layers that learn data representations at multiple levels of abstraction using artificial neural networks (ANN). Deep learning models have considerably improved state-of-the-art performance in several domains, such as data processing with high dimensionality, image detection, and so on [7]. The ANNs are trained to find complex structures in a dataset by using a backpropagation algorithm. The algorithm calculates errors made by the model during training, and the models' weights are updated in proportion to the error. The drawback of this learning method is that examples from both classes are treated the same. In that situation where the data is imbalanced and has overlapping challenges, the model will be adapted more to the majority class than the minority class, and difficult to learn from the overlap region, which can affect the performance of the models.

In chapter 6, a new deep learning-based method for predictive maintenance on ACMS data was proposed. The rationale behind the proposed method is to study the impact of highly imbalanced data with class overlapping using deep neural networks architectures. Also, to explore handling extremely imbalanced and class overlapping using the cost-sensitivity method in deep neural networks, which involves modifying the deep learning algorithm to favour both classes during model training.

As a result, using the ACMS dataset, a study was conducted to explore the influence of loss function on various deep learning architectures. Furthermore, the study proposed an improved loss function called rescale focal loss (RFL) to deal with the extremely imbalanced problem, while a new deep neural network architecture was created to deal with small class disjunct and class-overlapping problems that are inherent in the ACMS data. The following is the new derived RFL:

$$\mathrm{RFL}(p_{,t}) = -\,(1-(p_t))^{\gamma} log_{10}\,(p_t) * \theta_i \tag{8- 1}$$

Where θ_i is the logic weight of each class, (p,t)) represent the estimated probability of each class, and γ≥0 is the discount factor parameter that can be tuned for the best estimation.

The RFL is derived for deep neural networks, enabling the deep learning algorithms to respond favourably to both minority and majority groups and discount the small class disjunct during training. The new approach presents a unique way of changing loss function with respect to weights and a unique arrangement of neural networks; it also dynamically regulates the combined weight to produce a merged predicting result. The first experiment was conducted to test the effectiveness of the rescaled focal loss against other loss functions such as focal loss, Kullback Leibler divergence loss, hinge loss, cross-entropy loss. In order to maintain consistency in testing the RFL, an LSTM architecture was used, and the discount factor was set at γ = 0.5. the result indicated that RFL shows improved performance, especially for extreme imbalance cases as compared to other loss functions (chapter 6). It was observed that multiplying logic to the weight of each class in the RFL accounted for the performance improvement.

Second, an additional experiment was set out to determine the impact of an extremely imbalanced dataset on the various deep neural networks architectures. Also, in the implementation, an investigation was carried out to ascertain the impact of the rescaled loss function in conjunction with various network design architectures. The following network architecture ware considered, the deep bidirectional neural networks as compared to the unidirectional feedforward deep networks. A new network architecture was proposed known as the auto-encoder bidirectional gated recurrent network (AE-BGRU) to learn the relationships between variables in the ACMS data in the process overcome the challenge of class overlapping. The rationale behind the choice of method is based on the nature of the ACMS dataset (which is time-series based). Usually, time-series datasets are mainly trained using recurrent neural networks (RNN); the challenge with RNN's is that they suffer from vanishing gradient problems and has a short-term memory. Varnishing gradient problem arises when training a deep multi-layer RNN (feedforward network) with a gradient-based learning approach and backpropagation. In the process, the weight of each ANN is updated in proportion to the partial derivatives of the error function with respect to weight in each iteration [8]. The problem arises when useful gradient information is unable to propagate from the out layer back to the input layer of the model. In order to solve the vanishing gradient problem in RNN, the long-short term memory (LSTM) and gated recurrent unit (GRU) networks were developed to capture long time dependencies in the sequence learning and to handle the gradient vanishing problem through the use of modified hidden layers or gates, the elaborate explanation about the architecture of gated recurrent networks has been presented in Maren et al. [9] and Buda et al. [10]. This study did not investigate the impact of the vanishing gradient problem in RNN on the ACMS data. Instead, its focus is on exploring the effectiveness of RFL and extremely imbalanced datasets on the gated neural networks architectures, such as the GRU, which has been shown to handle the vanishing gradient problems intrinsically during training [11].

Due to the nature of the ACMS data (i.e., heterogeneous and time series in nature), time-series deep learning networks are chosen as baseline methods for the experiment. The following network architectures were considered unidirectional LSTM and GRU. Then a proposed network architecture utilizes the benefits of autoencoder (AE), Bidirectional gated recurrent unit (BGRU), and Convolutional neural networks (CNN) for effective learning., In the study, a unique network structure was explicitly designed and implemented for the imbalanced classification of ACMS data. In the network architecture, the core blocks are made up of BGRU, and each block contains a cell that stores information, the blocks comprise a reset and update gate, and the cells help in tackling the vanishing gradient problem, as shown in Janusz et al. [12]. The reset gate determines how to combine new input with previous memory, while the update gate defines how much of the previous memory to retain. BGR Units comprise two blocks. The input data is fed into the two networks, the feedforward, and feedback with respect to time, and both of them are connected to one output layer. The gates in bidirectional GRU are designed to store information longer in forward and backward directions, providing both the past and future context in a sequence, which enhances the learning relationship between variables, resulting in the model performance enhancement. The novel AE-BGRU network is designed to employ the RFL to minimise bias, the AE to detect failures, and the BGRU to forecast outcomes. The AE-BGRU model was trained using the imbalanced ACMS dataset. The suggested AE-BGRU architecture was compared against current algorithms such as a normal LSTM and BGRU (which uses a normal binary cross-entropy as a loss function). To ensure uniformity, the number of network layers in each architecture was kept constant.

The following conclusion was obtained as a result of the experiment: the AE-BGRU model performs better in terms of precision and recall, as shown in Table 6.3. In comparison to LSTM and GRU, the AE-BGRU model improves precision and recall by 25% and 14%, respectively. The new redesigned architecture and the loss function employed in AE-BGRU have resulted in improved performance. The following factors contribute to AE- BGRU's enhanced performance. An autoencoder helps in compressing the input variables into a reduced dimension space. The reduced latent variables with more promising features are used to train BGRU networks rather than the whole data. The latent variables are in the reduced form of the original data and make the AE-BGRU learn better with the reduced data. Moreso, because AE-BGRU uses a bidirectional learning approach, the input data is fed into the networks in two directions, the feedforward and feedback with respect to time, connected to one output layer. The gates in bidirectional GRU provide two ways to learn longer relationships between independent variables than the unidirectional feedforward networks that enhance the overall model performance. RFL helps control the bias, which improves the model's performance compared to the normal GRU and LSTM networks.

More study was carried out to investigate if adding CNN layers to the AE-BGRU network could aid in the learning of better correlations between variables. The results showed that performance had improved after the implantation. The following factors are attributed to the improved performance of the AE-CNN-BGRU model. First, if there is a correlation between the variables in a dataset (a process known as autocorrelation), BGRU or LSTM Networks account for the sequential dependency, and the networks treat all the variables as independent, ignoring any relationship that exists between both observed and latent variables. Whereas CNN uses a process known as convolution when determining a relationship between available variables in the dataset [11]. For example, in convolutional learning, given two functions ***f*** and ***g***, the convolution integral expresses how the shape of one function is modified by the other. Traditionally, CNNs were built to analyse multi-dimensional data, such as image classification, rather than account for sequential dependencies, as RNNs, LSTMs, and GRUs do [13]. The ability to use filters bank [14] to compute dilations between each cell, also known as "dilated convolution," is a key benefit of adding CNN layers for sequential learning. This allows the network layers in CNN to understand better the relationships between the different variables in the dataset, resulting in improved results. Finally, a challenge was encountered during the model training in the implementation. The AE-CNN-BGRU takes a mini-batch of the samples as input, and given that the dataset has an extreme imbalanced ratio, the batch samples are likely to contain fewer or non-samples from the positive class (component failure), the model will end up learning majority patterns of the negative class alone after running for some few epochs, most of the losses from the majority class will dominate the gradient; hence the learning algorithm would simply generate a trivial classifier that classifies every example as the majority class (Negative). The challenge was handled by using the weighted loss. After each mini-batch, the weights are updated in proposition to the number of samples in each class.

8.5. Deep Reinforcement Learning for predictive Maintenance Modelling

Deep reinforcement learning has been widely employed in a variety of applications, including but not limited to healthcare, computer vision, video games, natural language processing, finance, and education[15]. Despite the great potential benefits of deep reinforcement learning for increasing machine learning model performance, not much work was found in the open literature to explore its applicability for handling imbalanced classification in a time series dataset. Specifically the use of the ACMS dataset to train DRL algorithms for forecasting failures. As a result, a study was conducted to investigate if deep reinforcement learning can anticipate extremely rare events in aircraft predictive maintenance models. A proposed imbalanced learning algorithm based on the deep reinforcement learning approach is presented in Chapter 7. The proposed method investigates if deep reinforcement learning can be used to forecast exceedingly rare events, namely for predictive maintenance modelling.

Two new algorithms were designed, the Double Deep SARSA with a prioritized experience replay memory (DDSARSA+PER) and the double deep Q-network with experience replay memory (DDQN + PER). The result of the implementation shows that the application of deep reinforcement learning for extremely rare failure prediction is viable. Also, it was observed that the proposed algorithm outperformed the baseline DQN algorithm. Further analysis shows that DDSARSA performed better than DDQN using the ACMS data. DDSARSA was then further compared with previous implementations, the data-level resampling approach [6] presented in chapters 4 and 5, and cost-sensitive approaches in chapters 6 [16][17]. The overall results show that DDSARSA outperformed other approaches in terms of False Negative Rate (FNR) and False Positive Rate (FPR). The overall results show that DDSARSA outperformed other approaches in terms of False Negative Rate (FNR) and False Positive Rate (FPR). The average FNR for the DDSARSA+PER is approximately 0.05%, compared to that of cost-sensitive methods, which is approximately 15% FNR, and that of the data level method (SMOTE+RF), which is approximately 18%.

In contrast to a machine learning (regression or classification) model that predicts the probability of future outcomes, DRL utilises a reward function to optimise future rewards. The combination of the reward function, which helps to counter bias during model training, and the use of prioritised experience replay memory, which, instead of uniformly sampling transactions from replay memory, employs a prioritised approach; this also entails replaying the important transactions more frequently, which optimises the learning process, accounts for the significant performance improvement in DDSARSA. Deep neural networks' convolutions also aid in the learning of relationships between variables in the transactions. The improvement in performance was also aided by the eligibility trace. The influence of eligibility trace benefits the new algorithms by reinforcing complete sequences of actions from a single experience, which contributes to the proposed algorithms' increased performance. Because of their learning mechanism, specialised learning environment, and reward function, deep reinforcement learning systems can be claimed to be the best for imbalanced classification issues.

The proposed algorithm's impact on predictive maintenance: False alarms (FPR) in predictive maintenance systems can result in higher maintenance costs owing to unnecessary tests, whereas false positive (FPR) indicate that the model failed to forecast failure. A high FPR or FNR might potentially lower the level of trust in the equipment prognostics system. As a result, the goal is to bring both FNR and FPR down to an acceptable level. This implies the model will accurately identify fewer false alarms, lowering total operational costs and increasing vehicle availability and reliability. Figure 7-11 shows that the proposed DDSARASA model has a lower false-positive rate than existing techniques, implying that incorporating the DDSARSA model into the predictive maintenance system will anticipate infrequent failure with fewer false positives false negatives. The proposed DDSARSA key drawback is the use of a double deep neural network, which increases training time.

Finally, this research does not just propose a solution to the imbalance problem in aircraft maintenance by identifying gaps in the chosen sector or focusing primarily on the AIRMES project alone. It also provides a generic strategy for dealing with an unbalanced dataset in a range of application domains, especially when faced with the challenges of severe imbalance ratios, class overlapping, and small class disjunct. As a result, the findings of the study can be used to develop reliable and high-performance modelling in a number of industries.

8.6 Reference

1. Jiang H., Gupta MR. Bootstrapping for Batch Active Sampling. 2021; : 3086–3096.

2. Zhou ZH. Ensemble methods: Foundations and algorithms. Ensemble Methods: Foundations and Algorithms. 2012. 1–218 p. Available at: DOI:10.1201/b12207

3. Rofifah D. Pattern Classification Using Ensemble Methods. Paper Knowledge . Toward a Media History of Documents. 2020. 12–26 p.

4. López V., Fernández A., García S., Palade V., Herrera F. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences. Elsevier Inc.; 2013; 250: 113–141. Available at: DOI:10.1016/j.ins.2013.07.007

5. Hartati EP., Adiwijaya., Bijaksana MA. Handling imbalance data in churn prediction using combined SMOTE and RUS with bagging method. Journal of Physics: Conference Series. 2018. Available at: DOI:10.1088/1742-6596/971/1/012007

6. Dangut MD., Skaf Z., Jennions IK. An integrated machine learning model for aircraft components rare failure prognostics with log-based dataset. ISA Transactions. Elsevier Ltd; 2020; 113: 127–139. Available at: DOI:10.1016/j.isatra.2020.05.001

7. Pal B., Paul MK. A Gaussian mixture based boosted classification scheme for imbalanced and oversampled data. ECCE 2017 - Int. Conf. Electr. Comput. Commun. Eng. 2017; (1): 401–405. Available at: DOI:10.1109/ECACE.2017.7912938

8. Gamboa JCB. Deep Learning for Time-Series Analysis. 2017; Available at: DOI:arXiv:1701.01887 [cs.LG]

9. Dangut MD., Skaf Z., Jennions IK. Rare Failure Prediction Using an Integrated Auto-encoder and Bidirectional Gated Recurrent Unit Network. IFAC-PapersOnLine. Elsevier Ltd; 2020; 53(3): 276–282. Available at: DOI:10.1016/j.ifacol.2020.11.045

10. Buda M. A systematic study of the class imbalance problem in convolutional neural networks. 2017; Available at: http://www.nada.kth.se/%7B~%7Dann/exjobb/mateusz_buda.pdf

11. Johnson JM., Khoshgoftaar TM. Survey on deep learning with class imbalance. Journal of Big Data. Springer International Publishing; 2019; 6(1). Available at: DOI:10.1186/s40537-019-0192-5

12. Konar A. Artificial Intelligence and Soft Computing. Artificial Intelligence and Soft Computing. 1999. Available at: DOI:10.1201/9781420049138

13. Lecun Y., Bottou L., Bengio Y., Ha P. LeNet. Proceedings of the IEEE. 1998; (November): 1–46.

14. Lecun Y., Bengio Y., Hinton G. Deep learning. Nature. 2015; 521(7553): 436–444. Available at: DOI:10.1038/nature14539

15. François-lavet V., Henderson P., Islam R., Bellemare MG., François-lavet V., Pineau J., et al. An Introduction to Deep Reinforcement Learning. (arXiv:1811.12560v1 [cs.LG]) http://arxiv.org/abs/1811.12560. Foundations and trends in machine learning. 2018; II(3–4): 1–140. Available at: DOI:10.1561/2200000071.Vincent

16. David Dangut M., Skaf Z., Jennions I. Rescaled-LSTM for Predicting Aircraft Component Replacement

Under Imbalanced Dataset Constraint. 2020 Advances in Science and Engineering Technology International Conferences (ASET). IEEE; 2020. pp. 1–9. Available at: DOI:10.1109/ASET48392.2020.9118253

17. Dangut MD., Skaf Z., Jennions IK. Rare Failure Prediction Using an Integrated Auto-encoder and Bidirectional Gated Recurrent Unit Network. IFAC-PapersOnLine. Elsevier BV; 1 January 2020; 53(3): 276–282. Available at: DOI:10.1016/j.ifacol.2020.11.045

1. Jiang H., Gupta MR. Bootstrapping for Batch Active Sampling. 2021; : 3086–3096.

2. Zhou ZH. Ensemble methods: Foundations and algorithms. Ensemble Methods: Foundations and Algorithms. 2012. 1–218 p. Available at: DOI:10.1201/b12207

3. Rofifah D. Pattern Classification Using Ensemble Methods. Paper Knowledge . Toward a Media History of Documents. 2020. 12–26 p.

4. López V., Fernández A., García S., Palade V., Herrera F. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences. Elsevier Inc.; 2013; 250: 113–141. Available at: DOI:10.1016/j.ins.2013.07.007

5. Hartati EP., Adiwijaya., Bijaksana MA. Handling imbalance data in churn prediction using combined SMOTE and RUS with bagging method. Journal of Physics: Conference Series. 2018. Available at: DOI:10.1088/1742-6596/971/1/012007

6. Dangut MD., Skaf Z., Jennions IK. An integrated machine learning model for aircraft components rare failure prognostics with log-based dataset. ISA Transactions. Elsevier Ltd; 2020; 113: 127–139. Available at: DOI:10.1016/j.isatra.2020.05.001

7. Pal B., Paul MK. A Gaussian mixture based boosted classification scheme for imbalanced and oversampled data. ECCE 2017 - Int. Conf. Electr. Comput. Commun. Eng. 2017; (1): 401–405. Available at: DOI:10.1109/ECACE.2017.7912938

8. Gamboa JCB. Deep Learning for Time-Series Analysis. 2017; Available at: DOI:arXiv:1701.01887 [cs.LG]

9. Dangut MD., Skaf Z., Jennions IK. Rare Failure Prediction Using an Integrated Auto-encoder and Bidirectional Gated Recurrent Unit Network. IFAC-PapersOnLine. Elsevier Ltd; 2020; 53(3): 276–282. Available at: DOI:10.1016/j.ifacol.2020.11.045

10. Buda M. A systematic study of the class imbalance problem in convolutional neural networks. 2017; Available at: http://www.nada.kth.se/%7B~%7Dann/exjobb/mateusz_buda.pdf

11. Johnson JM., Khoshgoftaar TM. Survey on deep learning with class imbalance. Journal of Big Data. Springer International Publishing; 2019; 6(1). Available at: DOI:10.1186/s40537-019-0192-5

12. Konar A. Artificial Intelligence and Soft Computing. Artificial Intelligence and Soft Computing. 1999. Available at: DOI:10.1201/9781420049138

13. Lecun Y., Bottou L., Bengio Y., Ha P. LeNet. Proceedings of the IEEE. 1998; (November): 1–46.

14. Lecun Y., Bengio Y., Hinton G. Deep learning. Nature. 2015; 521(7553): 436–444. Available at: DOI:10.1038/nature14539

15. François-lavet V., Henderson P., Islam R., Bellemare MG., François-lavet V., Pineau J., et al. An Introduction to Deep Reinforcement Learning. (arXiv:1811.12560v1 [cs.LG]) http://arxiv.org/abs/1811.12560. Foundations and trends in machine learning. 2018; II(3–4): 1–140.

Available at: DOI:10.1561/2200000071.Vincent

16. David Dangut M., Skaf Z., Jennions I. Rescaled-LSTM for Predicting Aircraft Component Replacement Under Imbalanced Dataset Constraint. 2020 Advances in Science and Engineering Technology International Conferences (ASET). IEEE; 2020. pp. 1–9. Available at: DOI:10.1109/ASET48392.2020.9118253

17. Dangut MD., Skaf Z., Jennions IK. Rare Failure Prediction Using an Integrated Auto-encoder and Bidirectional Gated Recurrent Unit Network. IFAC-PapersOnLine. Elsevier BV; 1 January 2020; 53(3): 276–282. Available at: DOI:10.1016/j.ifacol.2020.11.045

CHAPTER 9: Conclusions and Further Research

This study provides four new data-driven predictive model techniques based on advanced imbalanced classification algorithms. An aircraft central maintenance system (ACMS) dataset and its accompanying maintenance records were utilised to validate the proposed algorithms. The proposed approaches concentrate on resolving some of the data irregularities identified in the ACMS data, such as an extreme data imbalance problem, irregular patterns and trends, class overlapping, and small class disjunct, which are major bottlenecks for traditional machine learning algorithms. The research's overall finding indicates that an advanced method for handling extremely imbalanced problems using the log-based ACMS datasets is viable for developing robust data-driven predictive maintenance models. Deep reinforcement learning (DRL) strategies, specifically the proposed deep reinforcement learning (the DDSARSA+PER model), outperform other methods in terms of false-positive rates when compared to the four implementations. The validation result suggests that the DDSARSA+PER model is capable of predicting around 90% of aircraft component replacements with a 0.005 false-positive rate in both A330 and A320 aircraft families studied in this research. When compared to machine learning algorithms that estimate the probability of future outcomes, the DRL algorithm performs better because of various characteristics, such as the reward function, which optimises future rewards. Second, instead of evenly sampling transactions from replay memory, prioritised experience replay memory adopts a prioritised approach; this also means replaying the essential transactions more frequently, optimising the learning process.

9.1 Addressing the Research Aim and Objectives

The research aim is to develop a data-driven predictive model for aerospace applications using advanced imbalanced classification algorithms. The aim is achieved by fulfilling the following objectives.

Objective 1:- To carry out a comprehensive literature review on the application of data analytics in aerospace, machine learning techniques and then investigate approaches and effects of imbalance problems in developing predictive modelling. To find the existing work and gaps and understand the shortcomings and underpin the countermeasures to be designed.

Evidence:- In chapter 3 and the review of related work sections of technical chapters (chapter 4 to 7), a review of imbalanced learning, application of data analytics in aerospace (including machine learning techniques) concentrating on the effects of rare failure prediction in aircraft maintenance was undertaken and presented. A review of the open literature supports the need for a more advanced method to deal with the current and future issues of handling severely imbalanced datasets with class overlapping and small class disjunct problems, particularly in aircraft maintenance modelling. In chapters 2, 3, 4, 5, and 7, the drawbacks of existing machine learning methods for addressing rare failure prediction or imbalanced classification problems and other research gaps were highlighted.

Objective 2:- To carry out data Pre-processing to quantify and understand various distribution and complexities inherent in the log-based aircraft ACMS datasets.

Evidence: In chapter 2, an exploratory data analysis is presented, which quantifies and explains various distribution and complexities inherent in the log-based aircraft ACMS datasets. In order to use log-based data to develop a robust data-driven predictive model, the first step is to interpret the logs, filter out a large amount of noise (that is, data irrelevant to the set goal) and extract predictive features. Also, the known failure cases need to be collected for learning and evaluation. The problem needs to be transformed into an appropriate learning scenario, and a performance measure that reflects real-world needs must be determined. The challenge in predicting rare failure using log-based time-series data is that the data distribution has irregular patterns and trends, affecting the learning of temporal features. Existing methods for handling slightly imbalanced datasets are understandable for certain types of data, such as image classification. However, it was unclear whether training extremely imbalanced, time-series data using the existing approach would improve model performance. Therefore, an important task that was carried out for this objective is to carry out exploratory data analysis and develop a novel algorithm to handle the pattern mining and transformation of the ACMS dataset for predictive modelling. The exploratory data analysis helps to see that the log messages (failure warnings) hold direct links to aircraft LRU failure, leading to replacement. It was also discovered that the target components are infrequent, causing the data to be imbalanced and overlapped.

Objective 3:- To design and develop a dynamic and robust Imbalance classification algorithm using machine learning, deep learning and deep reinforcement learning (DRL) strategies that will handle extreme class imbalance, class overlapping and class disjunct in both binary and multi-class scenarios.

Evidence:- In chapters 4, 5, 6, and 7, four unique algorithms based on pattern recognition, ensemble learning for managing class overlapping and rare minority problems, deep learning, and deep reinforcement learning-based models are proposed and implemented.

Objective 4:- To develop an aircraft predictive maintenance model and test it using the different testing datasets to establish its adaptability to various challenges.

Evidence:- Objective 3 and 4 works simultaneously. Therefore, implementing the four proposed algorithms for predictive maintenance modelling, handling the extremely imbalanced classification problem is performed inherently. The dataset used for training and testing the algorithms is collected from a fleet of sixty civil aircraft. The data comprises two databases: the operational failure log obtained from the aircraft central maintenance system (ACMS) and its corresponding maintenance records usually recorded by maintenance engineers (ground truth data). The two datasets are integrated and grouped according to the aircraft family. The two available aircraft families in the datasets are the A330 (22 aircraft) and (the A320 38 aircraft). Some components that are replaced due to unplanned maintenance are selected for validation in each family. The dataset has a data imbalance problem because of the rare representation of the target components, which the proposed technique seeks to address. After transforming the dataset in objective 2, the data was divided into two; 80% for model training while 20% for model testing. Chapters 4, 5, 6 and 7 present the algorithms for extremely imbalanced classification and aircraft' predictive maintenance model.

Objective 5:- To validate the model using ground truth data in order to ascertain its accuracy and performance.

Evidence:- validation was carried out using ground truth data available to ensure the quality of our proposed data-driven predictive model. The ground truth data is an actual maintenance record carried out by aircraft maintenance engineers. The models' predictive result was validated against the actual ground truth data (real failure leading to component replacement). These types of failures are infrequent, making the dataset highly imbalanced. Further analysis was performed to demonstrate the model's performance for predicting aircraft component failure within the desired time range, e.g. ability to predict a number of flights in advance of failure. It is important to make predictions within a reasonable period, not too far before the failure point (to avoid underutilising resources) and not too close to a failure point (to allow sufficient time to prepare maintenance action). Therefore, a reasonable prognostic period is taken between ten and two flights before a failure point (-not greater than ten flights and not less than two flights to failure).

9.2 Contribution to Knowledge

In the course of this research, a significant contribution to knowledge is recorded. This research has so far made the following contributions to knowledge.

1. This research has reviewed the literature that addresses the imbalanced learning problem across the academic and industrial sectors to understand the current research directions in predictive maintenance. The review of the literature contributes to knowledge by establishing familiarity with an understanding of current research in data analytics as it relates to aircraft' predictive maintenance. The review also confirms that the continued growth and availability of data on large-scale, increases more analytical challenges, such as the extremely imbalanced classification problem, class overlapping and other challenges related to class distribution that can cause performance degradation in machine learning models.

2. Expletory data analysis: The ACMS data analysis improved knowledge of variables by extracting averages, mean values, identifying trends by displaying data in graphs such as scatter plots and histograms, and discovering errors-outliers and missing values in the data. This information can be beneficial for future research and other studies that use comparable datasets.

3. Design and Implement an Algorithm for pattern identification and transformation: Developing a predictive maintenance model to predict unplanned failure of aircraft components using an imbalanced, heterogeneous and system log-based dataset is one of the significant contributions of this research. The dataset used contains extremely rare failures of the target component. A well-known natural language processing technique, the TF-IDF and vectorization are transformed and integrated for pattern identification and text vectorisation. Then an ensemble-based random forest algorithm was successfully adapted for individual functional item prediction. The algorithm can be used for pattern identification and classification in log-based datasets.

4. Developed an optimisation of ensemble learning-based algorithm for rare failure prediction: Another significant contribution is developing novel imbalanced learning implementation based on ensemble learning to handle the challenge of class overlapping. The new algorithm focuses on improving the detection of a rare failure in the log-based dataset. In addition, a hybrid framework for data-driven predictive maintenance was also proposed. The framework is based on a hybrid-ensemble method, which improves the prediction of the minority class during learning. The proposed Mix Gaussian Process with Expectation-maximization (MGP-EM) based algorithm computes the probability of points belonging to the cluster, which deals with an in-between point to avoid ambiguity problems in clustering. The proposed method overcomes the problem of class-overlapping or small-size samples, which is difficult for the classifier to learn, hence improving the prediction of a minority class. It also overcomes the problem of over-sampling in K-means clustering, which is sensitive to outliers and noise and unable to handle more massive datasets. The algorithms are robust and provide a high-performance solution for handling data imbalance problems, focusing on extreme imbalance ratio and irregular distribution (class drifting) in binary and multi-class contexts.

5. Developed a new approach for handling imbalanced datasets using a deep learning method. Training deep neural networks with an extremely imbalanced dataset, the overall total error cost representing the majority class usually overwhelms the minority by dominating the model's gradient, producing a bias model**.** A new method is proposed to address model biases in deep neural networks. The solution involves rescaling the loss function to respond favourably to the minority class during model training. The proposed techniques try to mitigate model biases using a derived re-scale loss function in neural networks. The re-scale loss controls the majority class's weights to balance with the weight of the minority class, hence enabling the model to respond favourably to both classes. The approach is tested using LSTM networks. Another deep learning implementation based on Autoencoder and bidirectional gated neural network (AE-CNN-BGRU) was proposed. The proposed approach first narrows down the volume of aircraft warning or failure messages into a small set of important and most relevant logs. It generates accurate link failure/warning messages in relation to aircraft LRU removals. The auto-encoder first trains the model with only negatively labelled data to detect rare faults using the reconstruction error threshold. The output of AE is used as input to the BGRU networks to predict those faults in Next-N-step. The evaluation indicates that AE-BGRU can effectively find the important log messages that hold direct links to aircraft LRU failure causes, leading to replacement.

6. Developed an imbalanced learning algorithm using a deep reinforcement learning approach for predicting extremely rare failure problems in complex aircraft systems.

The new deep reinforcement learning approach is designed to capture the patterns of extremely rare component failures adequately. The model is trained to predict aircraft component replacement well in advance of failure. The technique includes designing and developing an environment for the state-action, a reward function for rewarding agent-classifier actions, and the unique arrangement of a deep neural network architecture for policy optimization. The new method is validated using a real-world aircraft central maintenance system dataset. Exploring the ACMS dataset for developing a predictive maintenance model is a significant contribution because of its heterogeneous nature, challenging to analyse.

9.3 Intellectual Contribution and Impact

This research proposes novel methods and algorithms for rare failure prediction based on Machine Learning (ML), Ensemble learning, Deep Learning (DL), and Deep Reinforcement Learning (DRL), which provide new approaches to solving extremely imbalanced classification problems for rare failure event prediction in aircraft maintenance using aircraft operational log-based heterogeneous time-series dataset which is lacking in the literature.

This study also focuses on developing a predictive model for predicting aircraft component replacement with a unique capability of given prognostic alerts within a defined window. The model, when validated, can be integrated into the aircraft predictive maintenance system. Hence, reducing operational disruptions reduces the average delay time and improves aircraft utilisation, which will provide a cost-benefit to airlines.

Furthermore, this research has both industrial and academic impacts. The academic impact of this research comes through paper publications.

List publication can be found in: **https://orcid.org/0000-0003-2094-5370**

1. Dangut, Maren David, Zakwan Skaf, and Ian K. Jennions. "An integrated machine learning model for aircraft components rare failure prognostics with log-based dataset." ISA transactions 113 (2021): 127-139. DOI: 10.1016/j.isatra.2020.05.001

2. Dangut, Maren David, Zakwan Skaf, and Ian K. Jennions. “Rare Failure Prediction Using an Integrated Auto-encoder and Bidirectional Gated Recurrent Unit Network.” IFAC-PapersOnLine 53.3 (2020): 276-282. DOI: 10.1016/j.ifacol.2020.11.045

3 Dangut, Maren David, Zakwan Skaf, and Ian Jennions. "Aircraft predictive maintenance modeling using a hybrid imbalance learning approach." (2020). DOI: 10.2139/ssrn.3718065

4. Dangut, Maren David, Zakwan Skaf, and Ian Jennions. “Rescaled-LSTM for Predicting Aircraft Component Replacement Under Imbalanced Dataset Constraint.” 2020 Advances in Science and Engineering Technology International Conferences (ASET). IEEE. DOI: 10.1109/ASET48392.2020.9118253

5. Paper Under Review (the second step after minor correction): Dangut, Maren David, Zakwan Skaf, and Ian Jennions "Handling Imbalanced Data for Aircraft Predictive Maintenance using the BACHE Algorithm" Applied Soft Computing Journal (2021)).

6. Paper under review (the second step after minor correction): David Dangut, Ian Jennions, Steve King and Zakwan Skaf " Application of Deep Reinforcement Learning for Extremely Rare Failure Prediction in Aircraft" Journal of Mechanical Systems and Signal Processing (2021).

7. Paper under review (the second step after correction): David Dangut, Ian Jennions, Steve King and Zakwan Skaf "A Rare Failure Detection Model for Aircraft Predictive Maintenance Using Deep Hybrid Learning Approach" Journal Neural Computing and Applications (NCAA) (2021).

This research will also create an impact within the industry by contributing towards replacing unscheduled maintenance with systematic scheduled maintenance. This will help avoid aircraft operations disruption, reduce the average delay time and improve aircraft utilisation. In addition, the research does not only propose a solution for the imbalance problem in aerospace maintenance by identifying the gaps in the selected domain or focusing on the European AIRMES project alone. It also provides a generic solution, which could be implemented in different application domains faced with the imbalanced dataset - concentrates on the challenge of extreme imbalance ratio in the time-series big-data context. Therefore, the research outcome can generally be used in diverse industries to develop robust predictive maintenance modelling.

9.4 Limitation

One of the limitations of this study is only ACMS data and from one fleet is used for the validation.

9.5 Future Work

Several areas for further research have been identified at the course of this study, presented as follows:

1. This work can also be extended further by looking at the effect of class overlapping in the process of over-sampling the minority.

2.. In the future, we hope to develop this work further by looking at the effect of class overlapping in the process of over-sampling the minority class in the imbalanced learning context. We will also look at improving model performance by analysing the internal model structure to predict component replacement in the desired time window in advance -before failure to carry out actionable maintenance. Also, the BACHE algorithm can be developed as a python library to analyse severe imbalances in log-based datasets.

3. Further studies can be conducted on other architectures of AE-CNN-BGRU, such as transforming the time series into graphical representation using recurrence plots. The resulting images can be trained using CNN-BGRU for likely performance optimisation. Also, other aircraft data can be added to ACMS to enhance model training.

4. In the DRL approach, the work can be extended by carrying out further experimentation to determine the impact of high imbalanced on other deep reinforcement learning. Parameters such as changing the network architecture, an additional variable can be introduced into the deep neural network to keep track of the physical state and check for inconsistency with the physical laws to improve accuracy. Also, future work can consider enhancing performance optimization using other deep reinforcement learning algorithms. An ablation study will be carried out to assess the impact of eligibility trace and prioritise experience replay memory individually. More aircraft data sources - such as quick access recorder (QAR) Data, Performance Reports (PR), and Maintenance Tech Logs data can be integrated into the analysis. .

Appendices 1: Data Cleaning Process

[LR/SA]_ACMS_EXTRACT_FROM20060101_TO20160930.xlsx

LR = A330 and SA = A320

Maintenance data - SA & LR.xlsx

- In each file, the non-useful columns are removed for the proposed algorithms and merged both files into one by inserting the Maintenance data inside the ACMS messages.

- first two types of mistake correct:

1 FICTIVE FLIGHTS: When LEG N has the same ARRIVAL_AIRPORT as the DEPARTURE_AIRPORT of LEG N+2 (and are close in time). This implies that LEG N+1 lands where it takes off -> Inconsistent. If lines containing LEG N and N+2 are consecutive, or if there are some in-between lines with no DEPARTURE_AIRPORT and ARRIVAL_AIRPORT, the fictive LEG is removed (and the ACMS data, if there are, for LEG N+1 have their LEG updated to the previous or next LEG, depending on the closest EVENT_DATE).

2 SAME FLIGHTS: When LEG N and N+1 have the same DEPARTURE_AIRPORT and ARRIVAL_AIRPORT, and when their EVENT_DATEs are close enough in time, the two LEGs are considered as one as the same and merge them. This merge is only performed if the last line having LEG N and complete DEPARTURE_AIRPORT and ARRIVAL_AIRPORT, and the first line having LEG N+1 and complete DEPARTURE_AIRPORT and ARRIVAL_AIRPORT, are consecutive or contain in-between lines having neither DEPARTURE_AIRPORT nor ARRIVAL_AIRPORT information.

- Then care about having consistent information between EVENT_DATA and LEG_OF_OCCURENCE:

For an inputFile sorted by increasing EVENT_DATE, then LEG_OF_OCCURENCE, this script checks that the LEGs are increasing. If not, there is an inconsistency between dates and LEGs. The EVENT_DATE is modified in order the LEGs to be also increasing when it is sorted by EVENT_DATE.

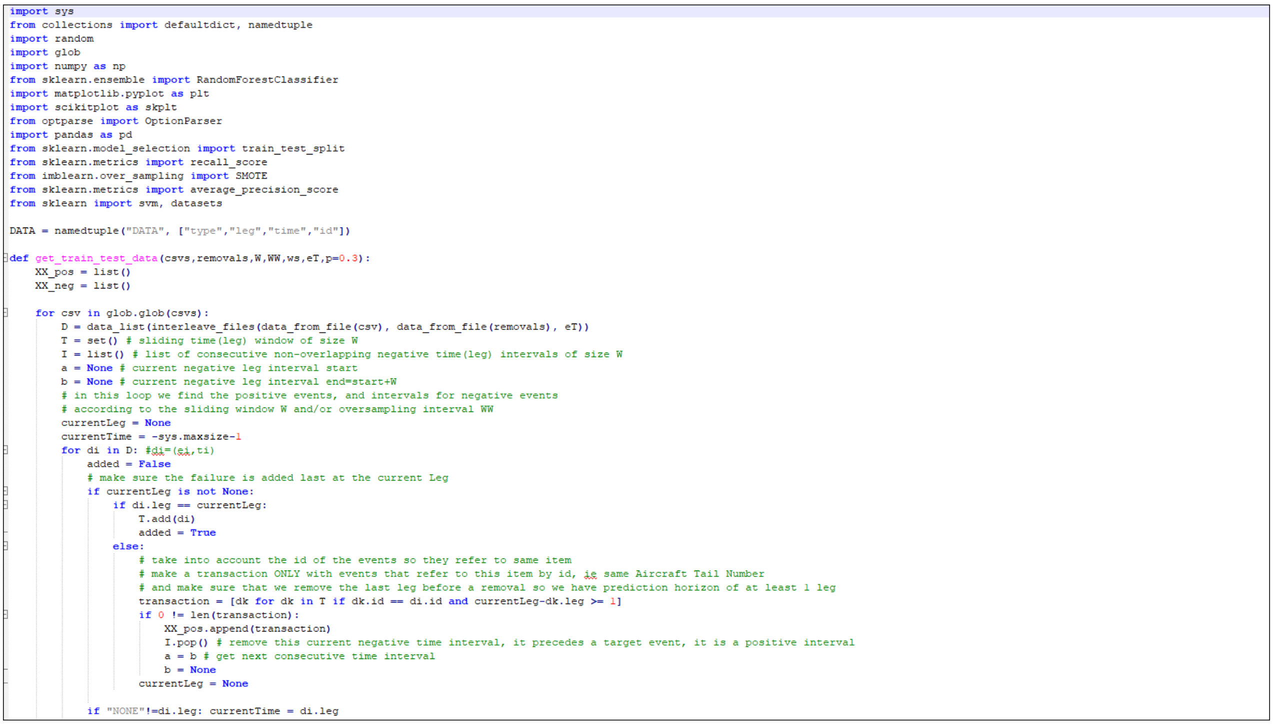
- The cleansed files (one for LR, one for SA) are called [LR/SA]_Datamergedgroupebytail_final.xls and contain corrected data, separated by tail (each tail has its own sheet), and is sorted by increasing EVENT_DATE and LEG_OF_OCCURENCE.

- Remark: the EVENT_DATE column is not taken in the ACMS data file because there is a bugg for flights landing the day after they took off. Instead, the FIRST_TRANSMISSION_DATE is considered. This means the information found in the EVENT_DATE column of our cleansed data files, for ACMS data lines, is actually the FIRST_TRANSMISSION_DATE.

Appendices 2: Project Codes

All the codes related to the projected can be found in an a GitHub https://github.com/dangutdavid/phd_codes_pdm. The data can be obtained based on request to email address: Maren.dangut@cranfield.ac.uk.

SMOTE with Random Forest Implementation



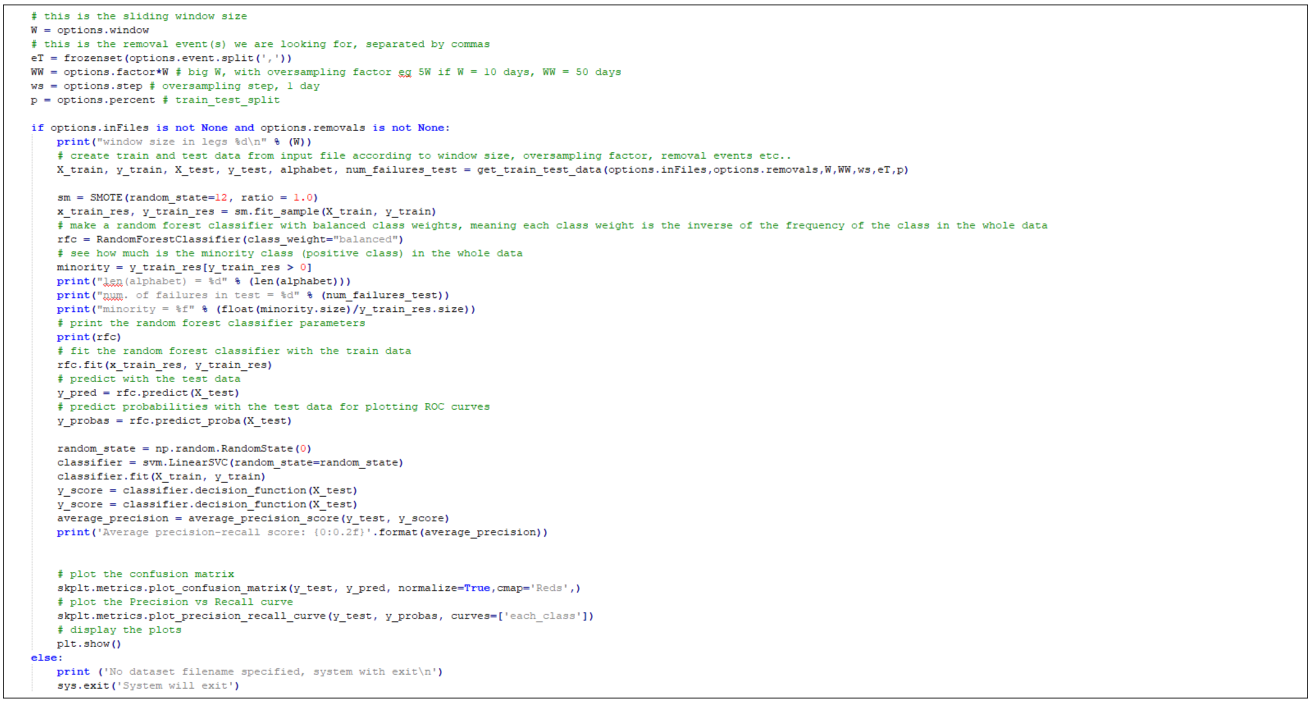
```
import sys
from collections import defaultdict, namedtuple
import random
import glob
import numpy as np
from sklearn.ensemble import RandomForestClassifier
import matplotlib.pyplot as plt
import scikitplot as skplt
from optparse import OptionParser
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import recall_score
from imblearn.over_sampling import SMOTE
from sklearn.metrics import average_precision_score
from sklearn import svm, datasets

DATA = namedtuple("DATA", ["type","leg","time","id"])

def get_train_test_data(csvs,removals,W,WW,ws,eT,p=0.3):
    XX_pos = list()
    XX_neg = list()

    for csv in glob.glob(csvs):
        D = data_list(interleave_files(data_from_file(csv), data_from_file(removals), eT))
        T = set() # sliding time(leg) window of size W
        I = list() # list of consecutive non-overlapping negative time(leg) intervals of size W
        a = None # current negative leg interval start
        b = None # current negative leg interval end=start+W
        # in this loop we find the positive events, and intervals for negative events
        # according to the sliding window W and/or oversampling interval WW
        currentLeg = None
        currentTime = -sys.maxsize-1
        for di in D: #di=(ei,ti)
            added = False
            # make sure the failure is added last at the current Leg
            if currentLeg is not None:
                if di.leg == currentLeg:
                    T.add(di)
                    added = True
                else:
                    # take into account the id of the events so they refer to same item
                    # make a transaction ONLY with events that refer to this item by id, ie same Aircraft Tail Number
                    # and make sure that we remove the last leg before a removal so we have prediction horizon of at least 1 leg
                    transaction = [dk for dk in T if dk.id == di.id and currentLeg-dk.leg >= 1]
                    if 0 != len(transaction):
                        XX_pos.append(transaction)
                        I.pop() # remove this current negative time interval, it precedes a target event, it is a positive interval
                        a = b # get next consecutive time interval
                        b = None
                    currentLeg = None

            if "NONE"!=di.leg: currentTime = di.leg
```



```
# this is the sliding window size
W = options.window
# this is the removal event(s) we are looking for, separated by commas
eT = frozenset(options.event.split(','))
WW = options.factor*W # big W, with oversampling factor eg 5W if W = 10 days, WW = 50 days
ws = options.step # oversampling step, 1 day
p = options.percent # train_test_split

if options.inFiles is not None and options.removals is not None:
    print("window size in legs %d\n" % (W))
    # create train and test data from input file according to window size, oversampling factor, removal events etc..
    X_train, y_train, X_test, y_test, alphabet, num_failures_test = get_train_test_data(options.inFiles,options.removals,W,WW,ws,eT,p)

    sm = SMOTE(random_state=12, ratio = 1.0)
    x_train_res, y_train_res = sm.fit_sample(X_train, y_train)
    # make a random forest classifier with balanced class weights, meaning each class weight is the inverse of the frequency of the class in the whole data
    rfc = RandomForestClassifier(class_weight="balanced")
    # see how much is the minority class (positive class) in the whole data
    minority = y_train_res[y_train_res > 0]
    print("len(alphabet) = %d" % (len(alphabet)))
    print("num. of failures in test = %d" % (num_failures_test))
    print("minority = %f" % (float(minority.size)/y_train_res.size))
    # print the random forest classifier parameters
    print(rfc)
    # fit the random forest classifier with the train data
    rfc.fit(x_train_res, y_train_res)
    # predict with the test data
    y_pred = rfc.predict(X_test)
    # predict probabilities with the test data for plotting ROC curves
    y_probas = rfc.predict_proba(X_test)

    random_state = np.random.RandomState(0)
    classifier = svm.LinearSVC(random_state=random_state)
    classifier.fit(X_train, y_train)
    y_score = classifier.decision_function(X_test)
    y_score = classifier.decision_function(X_test)
    average_precision = average_precision_score(y_test, y_score)
    print('Average precision-recall score: {0:0.2f}'.format(average_precision))


    # plot the confusion matrix
    skplt.metrics.plot_confusion_matrix(y_test, y_pred, normalize=True,cmap='Reds',)
    # plot the Precision vs Recall curve
    skplt.metrics.plot_precision_recall_curve(y_test, y_probas, curves=['each_class'])
    # display the plots
    plt.show()
else:
    print ('No dataset filename specified, system with exit\n')
    sys.exit('System will exit')
```