DOCTOR OF PHILOSOPHY

A novel feature selection approach towards predicting linear continuous movement on low powered devices

Hamid, Abdul

Award date:
2021

Awarding institution:
Coventry University

A Novel Feature Selection Approach Towards Predicting Linear Continuous Movement on Low Powered Devices

BY

ABDUL HAMID

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE UNIVERSITY'S REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

AUGUST 2020

Certificate of Ethical Approval

Applicant:

Abdul Hamid

Project Title:

A Novell Lightweight Machine Learning Approach for Predicting Linear Continuous Movement on Low Powered Devices

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval:

31 August 2020

Project Reference Number:

P109906

List of Publications

Eastwood, M., Konios, A., Tan, B., Jing, Y., & Hamid, A. (Accepted/In press). Conditional Random Field Feature Generation of Smart Home Sensor Data using Random Forests. In 2019 International Microwave Biomedical Conference (IMBioC 2019) (Vol. (In-press), pp. (In-press)). IEEE.

Jing, Y., Eastwood, M., Tan, B., Konios, A., Hamid, A., & Collinson, M. (Accepted/In press). An Intelligent Well-being Monitoring System for Residents in Extra Care Homes. Paper presented at International Conference on Internet of Things and Machine Learning, Liverpool, United Kingdom (2017).

Abdul Hamid, Michael J. Duncan, Emma L. J. Eyre & Yanguo Jing (2020) Predicting children's energy expenditure during physical activity using deep learning and wearable sensor data, European Journal of Sport Science, DOI: 10.1080/17461391.2020.1789749

Acknowledgements

First and foremost, I would like to convey my deep and sincere gratitude to my principal supervisor, Dr Yanguo Jing, for inspiring me to advance my career and providing me with this opportunity of self-development. His patience, motivational guidance and unremitting support during my academic journey have been instrumental in my personal development. My solemn thanks must also go to my second supervisor Dr Xin Lu for his guidance, advice and installing confidence in my abilities to solve various scientific and technical challenges.

Secondly, I would like to thank Prof Michael Duncan, Dr Mark Elshaw and Prof Kuo-Ming Chao, for their time and efforts in sharing their expertise, constructive critiques and insights that have helped steer my research in the right direction.

I would like to also profoundly thank my parents for their encouragement without which I would not have even dreamed of perusing a PhD. And finally, my wife and two sons for their continued patience, support and understanding through the exhaustive last few years.

Abstract

Machine Learning (ML) on low powered devices is still in its infancy, and the use of ML for continuous linear movement analysis is limited. There is a need to find an effective way to reduce the amount of computational load on low power devices. One of the ways to do so is to reduce the number of features within the ML model. This research took on that challenge and proposed a novel feature reduction approach by automatically selecting suitable and relevant feature subsets that enable ML models to achieve acceptable performance when deployed to low powered devices.

At the core of this thesis, it focuses on the methodologies that can help to reduce the complexities of the machine learning model to improve its efficiencies and enable it to operate successfully on low powered devices with limited computational resources. Internet of things (IoT), wearable devices and data-driven techniques offer the ability for practitioners to collect vast volumes of data and process them promptly drawing useful insights. At the same time, while the information is still helpful, practitioners can propose suitable interventions based on the information extracted by the physical activity monitoring device.

The need for Lightweight ML is because the objective of this research is to deploy a suitable solution on low powered devices that can effectively operate within the limited computational resources available. This thesis has successfully proposed, developed and tested a novel lightweight ML approach for linear machine learning problems. The proposed method has also achieved its aims of automatically selecting appropriate features for machine learning applications. The thesis shows how the novel technique can significantly improve computational efficiency by exploiting the correlation co-efficient and variance between elements to eliminate irrelevant features. This method has been piloted and tested in one publicly available dataset and two case studies – a study of children's physical activities and energy expenditure, and a swim classification study. These case studies demonstrate that the proposed novel approach is useful in selecting appropriate features and reducing model complexity while performing physical activity recognition and monitoring. The thesis also demonstrated the effectiveness of deploying the lightweight ML method on low powered devices by significantly increasing the computational efficiency. The approach was

evaluated using several supervised machine learning algorithms including; decisions trees, linear & logistic regression, random forests, multilayer perceptron, support vector machines.

The result of the "children's physical activity" case study has shown that this newly proposed approach can effectively predict children's energy expenditure and perform well within 5% of the baseline model. The best performing model using boosted trees achieved a 90% predictive accuracy inline with other research. The research has also shown that it can significantly increase the model's computational efficiency by up to 75% enabling it to operate on low powered devices with limited computational resources successfully. The "swim classification" case study explored the novel lightweight model's performance on physical activity classification for swimming. The best performing model showed that the novel approach effectively reduced the feature complexity and model dimensionally to achieve 78% classification accuracy while being within 5% of the baseline and gaining above 70% computational efficiency running on the low powered device (a device limited in computational resources). This novel lightweight approach can significantly reduce the models feature dimensionality enabling suitable ML algorithms to operate on low powered devices, saving time and computational resources. It has been incredibly effective at dealing for linear tri-axial time-series based Physical Activity monitoring and recognition problems.

Although this proposed approach was specifically proposed for linear machine learning problems, the thesis has further explored this novel approach in a non-linear, time-series, activity-based situation in a "smart care home" case study. As anticipated, the experiment has found that this novel approach was unsuitable and ineffective, and an alternative technique using Conditional Random Fields was recommended for future expansion of the approach to tackle datasets non-linear in nature. Finally, the contributions to the body of knowledge from this thesis and future directions of work have been provided.

Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Background & Motivations

A sedentary lifestyle is one of the major causes of severe health problems (Dijkhuis et al., 2018). Researchers have produced evidence of regular Physical Activity (PA) helping to reduce the risk of a variety of health problems significantly, such as substantially reducing the risks associated with cancer (Murray et al., 2020), the treatment of asthma (Endre & DuBuske, 2018) and supporting injury recovery (Ekegren et al., 2020). Based on the compelling evidence in support of promoting regular PA, there is a need for monitoring and measuring suitable levels of PA activity is being performed. As over excessive PA can lead to unphysiological loads on the joints (Greca et al., 2019). Wearable devices have created a particular use case for researchers to monitor PA. Repetitive weight-bearing PA's such as walking, cycling and jumping activities are recommended for children to promote bone health (Greca et al., 2019; Landry & Driscoll, 2012) which can be monitored and measured using wearable devices. Swimming, another repetitive PA, is a complementary therapy for people living with asthma (Endre & DuBuske, 2018) can also be tracked and measured using suitable waterproof wearables. Embedded sensors and wearables technology are increasingly playing an active role in healthcare-related research enabling experiments to take place safely outside of laboratories (Brodie et al., 2018; Dennison et al., 2013). Internet of things (IoT), wearable devices and data-driven techniques offer the ability for practitioners to collect vast volumes of data and process them promptly drawing useful insights. At the same time, while the information is still helpful. Practitioners can propose suitable interventions for individuals to promote better health (Dijkhuis et al., 2018). This research benefits both practitioners and those who are health conscious for monitoring and maintaining a healthy, active lifestyle.

Traditionally organisations have used data retrospectively to understand why current events have occurred and adjusted their practices only afterwards. More and more organisations are realising the benefits of predictive data analytics and use data prospectively (Nisbet et al., 2018b). By anticipating behaviour and automating prescriptive decision-making, organisations can manage the allocation of resources better and grow their business while minimising risk (Nisbet et al., 2018b). The process can also help to generate new digital information, assets that can add value and enable organisations to be more proactive in taking action (Kavakiotis et al., 2017).

The term Predictive Analytics generally describes the process of predictive modelling, forecasting based on meaningful probabilities modelled on descriptive analytics (Kotu & Deshpande, 2015). The term may also be used interchangeably with data mining (Nisbet et al., 2018b); however, it is more closely associated with machine learning (Kotu & Deshpande, 2015). Predictive machine learning algorithms take patterns discovered from within current and historical data to model and predict future trends (Zhou et al., 2017). Predictive Analytics has evolved from data mining techniques to offer a more unobstructed view of what is happening now and what is about to happen next (Finlay, 2014).

The application of Machine Learning (ML) is having a significant impact on analysing patterns from the large volumes of data produced by wearable devices (Saez et al., 2017). Suitable ML techniques are essential for the precise measurement of PA (Farrahi et al., 2019). However, organisations face many challenges when investing in machine learning technologies for analysing wearable sensor data (Papagiannaki et al., 2019). With the growing acceptance of ubiquitous use of microdevices and wearables, there is a real need for providing beneficial results in near real-time effectively (Rahmani et al., 2018). Firstly, performing the analysis in near real-time has always been difficult based on the availability of appropriate computing resources to compute the sensor readings where there may not be internet connectivity such as indoor swimming pools. Secondly, while relying on cloud-based solutions and internet connectivity, it limits the ability to deliver timely results (H. Cai et al., 2017). Finally, there are mounting privacy and security concerns (Xiaoliang Wang et al., 2019). Fears manifest as many solutions leverage cloud-based computing to analyse sensor readings and potentially sensitive data before providing any useful insights (Rajput et al., 2020). Cloud-based solutions, can in some cases, store data indefinitely which may also be difficult to trace as to who has access to the data as several vendors may be involved such as Google, Microsoft and Amazon. These vendors all offer cloud-based machine learning services which other organisations can leverage to help analyse their large volumes of heterogeneous data and offer their customers useful insights. With the current state-of-the-art cloud-based solutions such as Google's TensorFlow and Alexnet, modern architecture has become ever more complicated with a narrow focus on improving accuracy and without appropriately addressing computational efficiency (Xiaying Wang et al., 2020).

Further research focusing on reducing the computational time by reducing the model size has become necessary (Jacob et al., 2018). These challenges have created an urgent need for a solution that can provide near real-time results for monitoring PA while addressing these challenges by better leveraging Edge and Fog Computing (Greco et al., 2020). The terms Edge and Fog Computing refer to the concept of utilising the computational and storage resources, which are usually limited, closest to where it's needed, such as microdevices and wearables where the sensor readings originate (Escamilla-Ambrosio et al., 2018).

Another challenge when attempting to use an ML pipeline which has a subset of a much larger ML workflow such that practitioners are required to apply their knowledge in several areas, namely: data-pre-processing, feature engineering and feature selection before being able to model the data (Wever et al., 2018). Furthermore, practitioners must also appropriately choose an algorithm and optimise parameters to be able to accurately extract useful patterns and insights (Nithya & Ilango, 2017). These steps require considerable amounts of time to effectively achieve acceptable accuracy and hence creates a barrier for businesses to adopt the technology (Waring et al., 2020).

Research shows that in many IoT applications which attempt to push data processing to the low powered device can distribute the computation on separate layers (Samie, Bauer, et al., 2019). The initial data preprocessing can be performed directly on the device layer, and the intermediate result sent over to the fog layer for further processing (Samie, Tsoutsouras, et al., 2019). Several published articles present evidence of successful implementations of low powered devices performing well while modelling datasets for both classification and regression problems that span from speech synthesis, intrusion detection, augmented reality to PA (J. Chen & Ran, 2019). ML algorithms include, but not limited to, K-Nearest Neighbors (k-NN), Support Vector Machines, Decision Trees (DT) and or Artificial Neural Networks (ANN) (Sakr et al., 2020). However, attempts made in applying ML models to PA has traditionally relied on high-performance devices (Agarwal & Alam, 2020; Yao et al., 2017).

There is a need for a lightweight ML approach that can run on a low powered device that can produce near real-time results while also addressing the data security and privacy concerns. Near real-time has been defined in this approach as results produced within 8 seconds of the model receiving its input signal.

Some research has been done in this area so far. However, the current research challenges are:

1. Lack of an efficient, lightweight ML approach that produces results in near real-time.
2. There is no lightweight ML research addressing PA activity running on low powered devices.
3. There is limited literature on ML approaches designed to run on low powered devices.
4. Lack of automatic feature selection for lightweight ML applications running on low powered devices.

1.2 Aims & Objectives

The thesis aims to develop a novel lightweight automatic feature selection approach running on a low powered device, which is limited in computational resource, to produce results in near real-time for continuous linear physical activities. A continuous repetitive physical activity in a straight line such as running, jogging, walking, swimming, and cycling is considered a linear physical activity which produces a consistent signal. The thesis will demonstrate the use of the lightweight approach running on a device with computationally limited resources while preserving data privacy and producing accurate results within 5% of the baseline model.

Figure 1. The research focus in a typical automated ML workflow

Figure 1 describes the research focus, where it's believed that this thesis will have the most impact in solving the challenges in delivering timely results.

The objectives of the thesis are listed below:

1. Carry out a comprehensive review of lightweight ML applications for deploying on low powered devices.

2. Develop a novel automatic feature selection approach that can be used for lightweight ML applications deployed on low powered devices for continuous linear repetitive physical activity.

3. Evaluate the proposed new approach in two separate case studies in physical activity monitoring and recognition.

4. Explore other areas of potential use cases that can benefit from the research and make recommendations for future work.

1.3 Thesis Key Topics

The key topics outlined within the thesis, as depicted in ***Figure 2***, shows the main steps taken towards addressing the aims and objectives. The work-focused around better understanding the computational approaches that are currently in use or being examined to aid researchers in improving the effectiveness of feature selection optimisation. In particular, the focus within this thesis has been on developing a novel approach, using ensemble machine learning techniques in sequential modelling, for feature selection optimisation by understanding relations within the data and that which is especially useful in activity-based recognition.

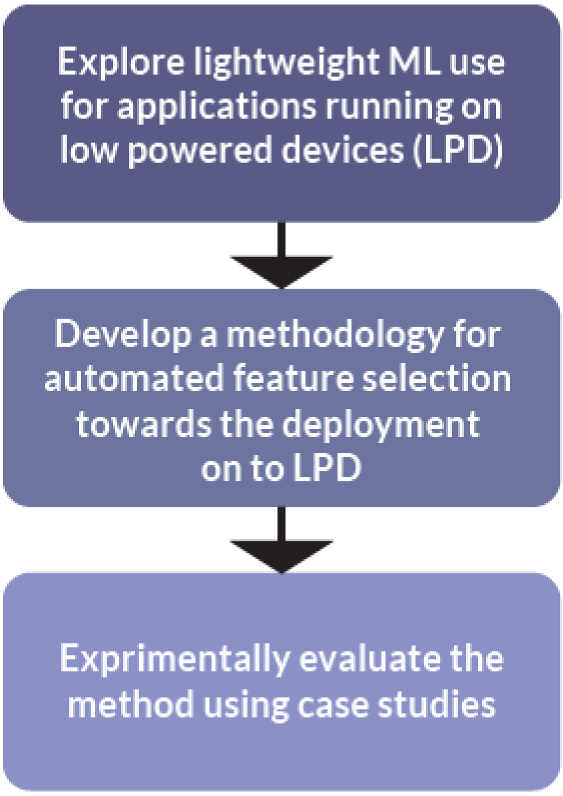


Figure 2. The key topics within the Thesis

1.4 Contributions

The thesis presents empirical comparisons of the method using two case studies. It demonstrates its significance in automated feature selection, several machine learning problems; including binary classification, multi-class classification and regression problems.

1. A novel, lightweight ML approach has been proposed to run effectively on lower power devices. This novel method has been validated using several supervised ML algorithms including decisions trees, linear & logistic regression, random forests, multilayer perceptron and support vector machines and using various datasets of numerous complexities.

2. An automated feature selection method was developed to extract the relevant features to reduce the feature dimensionality and increase the model's computational efficiency so that the ML model can effectively run on lower-powered devices.

3. The ML approach was first applied to a case study where supervised machine learning is used to accurately approximate children's energy expenditure or Metabolic Equivalents (MET). Body movement data is collected using wearable devices worn by children with an emphasis on physical activities that represent locomotor and object control movements commonly undertaken by children. The thesis discusses and compares the effectiveness of this method in **Chapter 5**. The findings showed that the technique was able to score within 5% of the baseline model while also improving the efficiency of the ML model by up to 70%, which is ideal for running on low powered devices.

4. This ML approach has also been evaluated in a case study where supervised machine learning is used in swim activity detection. The supervised machine learning methods used in this case study were Linear Regression, LSTM, kNN, Random Forests, Support Vector Machines, Boosted Trees and Neural Networks, the novel feature selection method successfully identified suitable features to train them. The thesis discusses and compares the effectiveness of this method in **Chapter 6**. Machine learning models such as Random

Forest, CNN, and Boosted Trees all performed above 70% accuracy in predicting energy expenditure. The computational efficiency gains were above 45% across all the models evaluated and up to 70% using the classical feed-forward artificial neural network.

5. The thesis has further explored whether the proposed novel lightweight ML approach could be suitable for non-linear based activities; it was evaluated using the CASAS smart homes activity recognition dataset. As anticipated, the study has shown that the novel light ML approach was ineffective at appropriately classifying the activities to an acceptable level of performance. Instead, an alternative approach was recommended using a non-parametric method based on the initial experimental results.

1.5 Thesis Structure

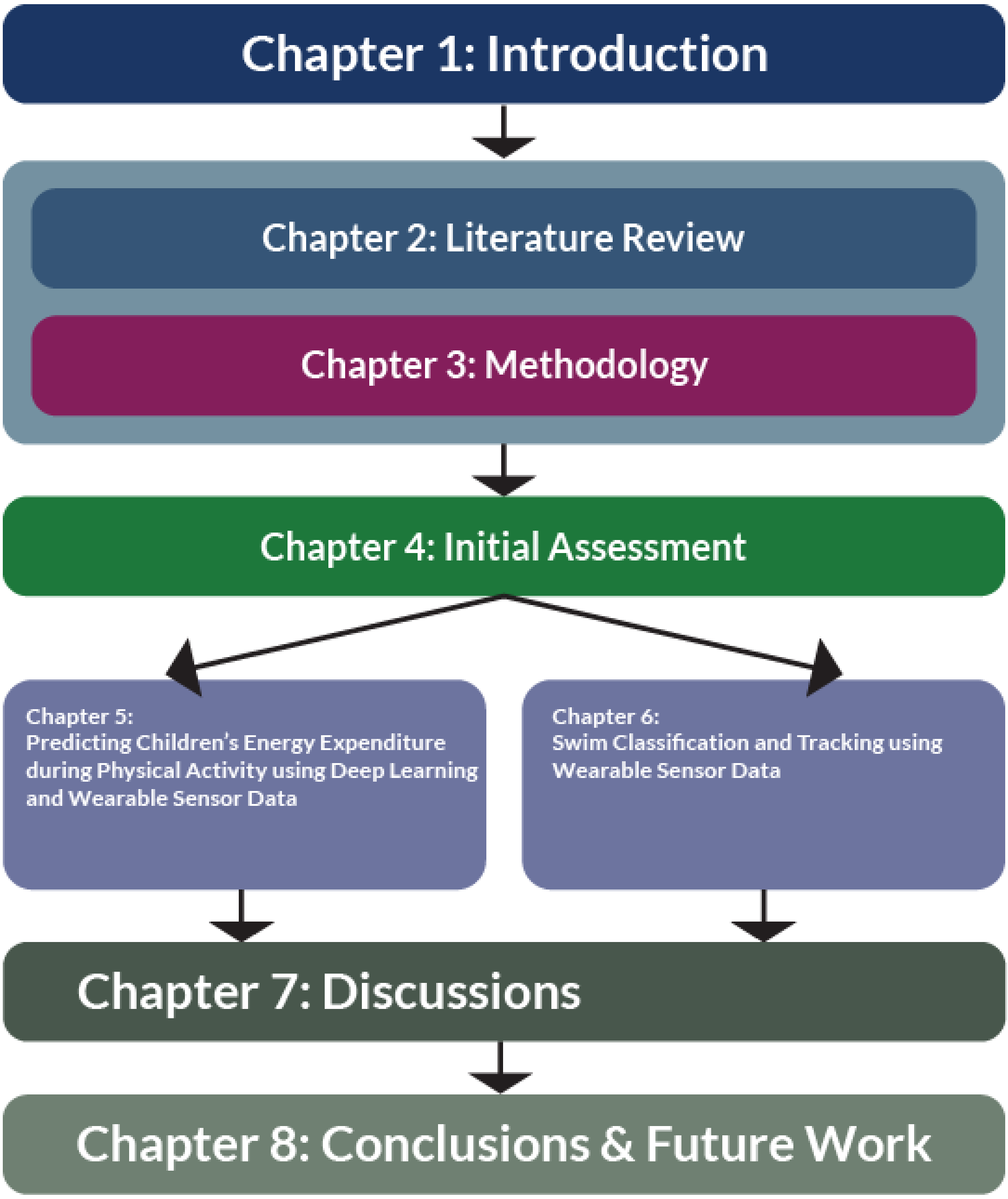


Figure 3. The Thesis Structure

The structure of this thesis is depicted in ***Figure 3***. **Chapter 2** provides a comprehensive literature review and outlines the fundamental importance of appropriate feature selection and its impact on machine learning models' performance. The performance issue mentioned here refers to data pre-processing, including feature selection as part of the ML process. The proposed novel feature selection method is described in **Chapter 3**. Subsection 3.1 explains some of the core components of the proposed method that has been designed to understand linear repetitive activity-based data to select suitable features. The approach's objective is to reduce the number of features for improving data modelling efficiency while also being useful in deploying it on low powered devices. The chapter describes how the new approach is considered efficient and lightweight. It aims to reduce the model's complexity by building a reduced feature subset by evaluating the correlations between features.

Chapter 4 presents the initial findings of the method's effectiveness in selecting suitable features using open source automated supervised machine learning pipelines using three types of machine learning problems. The results showed that the technique was able to score within 5% of the baseline methods while also improving the ML model's efficiency by up to 70%, which is ideal for running ML models on low powered devices. The experiments also revealed that the performance gains were only prevalent on those datasets with large sample sizes where correlations were present between features.

Chapter 5 presents one of three experimental case studies that demonstrate the effective use of the proposed feature selection method in several supervised machine learning techniques which showed that the energy expenditure could be predicted using the accelerometer readings from a wearable device. The study results were consistent with prior work undertaken in adults by Montoye, Westgate, et al. (2018), which reported that machine learning models predicted physical activities with an accuracy of 71-92% from wrist-worn GENEActiv accelerometers. Machine learning models such as Random Forest, CNN, and Boosted Trees all performed above 90% accuracy in predicting energy expenditure. The computational efficiency gains were above 50% across all the models evaluated and up to 70% using Boosted Trees.

Chapter 6 presents the final case study in demonstrating the effectiveness of the proposed feature selection method in selecting the suitable features for supervised machine learning models in swim activity recognition on the lower-powered device. The proposed method was able to select relevant features to train a neural network model that achieved 73% lap (begin and end of swim laps) accuracy and was able to identify the swim stroke type at 78% accuracy. The results showed that computational efficiency gains of 45% were achievable, and above 70% was possible using the shallow MLP model, which achieved a score within 5% of the baseline model.

In **Chapter 7**, the summary of findings from the experimental case study is outlined and discussed. The chapter highlights and explains how effective the approach has been in tackling linear time-series based physical activity recognition and an attempt is described at tackling a non-linear activity recognition problem using the CASAS smart home dataset. As expected, the results indicated the approach to be ineffective and instead, an alternative solution was recommended based on the initial experimental results.

In **Chapter 8**, the summary thesis conclusions, discussions and future works are described.

Chapter 2: Literature Review

2.1 Related Work

The connected devices which constitute the Internet of Things (IoT) are generating huge volumes of data (Yousefpour et al., 2019). Yousefpour et al. (2019) indicated that this surge in traffic would negatively impact on the mobile data network architecture and have trouble coping with the increasing load. The authors also mention that there are specific low latency applications that cannot operate using cloud-based services. The edge computing paradigm, which refers to the concept of providing compute and storage resources to the edge of the network and aims to solve these challenges (Yousefpour et al., 2019), has drawn more and more attention in recent years. For instance, edge computing is a crucial component in the current landscape for augmenting edge (IoT) devices (W. Chen et al., 2019). It integrates low powered devices with cloud services by filtering, pre-processing, and aggregating data deployed close to the edge (De Donno et al., 2019).

Furthermore, while there are many advantages of expanding the use of IoT devices, there are implications of cloud supported IoT applications around rising concerns over security and privacy risks (Ni et al., 2018). Ni et al. (2018) explore state-of-the-art solutions to address security and privacy issues such as differential privacy and fully homomorphic encryption. However, the authors mentioned that these techniques are still in their infancy and don't necessarily solve latency issues. More and more research focuses on running data analysis or data processing on low powered devices rather than sending all the data to the cloud for processing to solve the latency and privacy issues. The increasing interest in deploying ML models on edge devices creates a need for further exploration of edge computing techniques such that ML applications can effectively operate on low power devices while also eliminating latency issues and reducing privacy concerns.

2.1.1 Feature Selection on Low Powered Devices

Selecting the most appropriate features from a given dataset is an essential part of a typical ML workflow. The ML pipeline is a subset of a much larger ML workflow such that practitioners are required to apply their knowledge in data-pre-processing, feature engineering and feature selection to make the dataset useful to ML models. Furthermore, practitioners must also appropriately choose an algorithm and optimise its parameters. These steps require considerable amounts of time to achieve acceptable accuracy effectively and, hence, create a barrier for adopting suitable ML models for low-powered devices.

There is a consensus that the most time-consuming portion of the ML workflow is data preparation (Elkan, 2010; Zhou et al., 2017). A chosen ML model's performance depends mainly on the algorithm and the features selected from the dataset (Uysal, 2016; Zhou et al., 2017). In other words, just feeding an ML model with raw data is not enough to achieve good results. Appropriate feature selection is essential when dealing with extremely high dimensional datasets which have created a further challenge as ML models tend to suffer from the problem of overfitting (Höglund, 2017). The accumulation of vast volumes of data creates complex data structures. The dataset's increased complexity directly affects traditional hypothesis-driven statistical approaches such as linear regression, logistic regression, and or stochastic modelling techniques that have limited capabilities in processing information efficiently from large complex datasets (Elkan, 2010). Therefore, there is an urgent need to develop advanced, efficient solutions that can help to manage high-dimensional datasets effectively on devices with limited computational resources (Inoubli et al., 2018).

The extraction of new discriminative features from raw data using domain knowledge to improve ML algorithms' performance is known as feature engineering (Davis & Foo, 2016). Feature engineering is an essential step in the ML pipeline. The features often determine the effectiveness of the model more so than the chosen algorithm (Dipanjan et al., 2018). Feature engineering is traditionally an iterative process, which requires manual intervention and involves several tasks such as removing irrelevant data, removing missing values, transforming and or aggregating values. This iterative process allows features to be evaluated against ML performance until the desired accuracy is achieved. The process is usually very time consuming, and labour intensive that automated feature engineering aims to improve. This essential step compounds in complexity, even more, when attempting to analyse real-time data using deep learning techniques deployed on low powered devices (J. Chen & Ran, 2019). J. Chen & Ran (2019) provide a comprehensive review of the current state-of-the-art deep learning techniques employed for edge computing uses. The authors focus on the need for viable solutions for dealing with low-latency and high computational challenges faced by researchers when dealing with ML models running on low powered devices.

By improving this process's efficiency by automatically generating new discriminative features that are better for training ML models (Davis & Foo, 2016) could potentially help to deal with the high computational challenges for edge computing described by J. Chen & Ran (2019). Yao et al., (2017) introduce DeepSense, a deep learning framework that deals with feature customisation challenges in a unified manner for time-series mobile sensor data processing a low powered device, Nexus 5 and Intel Edison. The authors also described how exploiting the local interactions among the sensor readings effectively extracts temporal relationships between features to model the data on low-powered devices accurately.

Davis and Foo, (2016) outline several automated feature engineering techniques that are commonly used such as Boolean operators, M-of-N, X-of-N, hyperplanes, standard arithmetic operators, Bayesian networks and domain knowledge annotation methods. The authors proposed approach successfully derived suitable relevant features using their automatic feature engineering library. Their evaluation concluded that their ML classifier performed better on their test data and proved that automated feature engineering could be successfully applied to other classification problems. (Kanter & Veeramachaneni, 2015) introduced Deep Feature Synthesis, an algorithm designed to generate features for relational datasets automatically. The authors demonstrated how new features could be automatically generated by following relationships in data using three datasets from different domains to evaluate their algorithm's effectiveness. Feature engineering can drastically improve ML algorithms' performance; however, it can increase the data model's complexity, reducing its efficiency. For example, automatic ML tools are great at extracting large sets of simple statistical features from the data, however, at the costs of increasing the high dimensional feature space which can render the model inefficient (Leppänen et al., 2020). Therefore, a balance is required to handle continuous real-time time-series data and maintain the performance while operating on the low-powered device effectively (Shafiq et al., 2020).

The goal of feature selection is to improve the adequate performance of any ML model while also increasing computational efficiency for deploying on low-powered devices; therefore, it is essential to establish the key variables necessary to achieve acceptable results. Data preparation is a crucial step in a typical ML pipeline as raw data is very rarely in the correct format and is usually unstructured, inconsistent and or incomplete (Zhou et al., 2017). For most ML algorithms, data is typically transformed into a matrix which is used as the input. During this process, data is cleaned, transposed and in most cases, its dimensionality reduced. The data attributes are not all equally important or indeed useful to ML algorithms' performance, thus decreasing its dimensionality can help reduce the complexity of the data model improving its efficiency (Nisbet et al., 2018a). The process of lowering the number of features, computationally, based on relevance and selecting only those optimal features for input to ML algorithms is termed feature selection (J. Cai et al., 2018; Molina et al., 2002). Other advantages include avoiding overfitting, reducing storage needs, reducing training time (Zeng et al., 2015), and better understanding the processes that generated the features (Guyon & Elisseeff, 2003). The more training samples of the given dataset there are the better an ML algorithm can perform; however; computation time will also increase proportionally. Several methods exist in feature selection, and they fall in either one of three categories as described by Sheikhpour et al., (2017): supervised, unsupervised and semi-supervised feature selection. Hindawi et al. (2013) explained that a feature's relevance might be different depending on the learning method used to determine its importance.

Chandrashekar et al., (2014) presented a generalised overview of commonly used feature selection techniques in supervised learning and focused on Filter, Wrapper and Embedded methods. They found that the more information you extract using feature selection methods does not always increase the performance of the machine learning model. However, after experimentation using seven different datasets using two classifiers, they found that feature selection can provide benefits such as providing insight into the data, improve the classifier model’s performance and enhance variable generalisation and elimination.

Sheikhpour et al. (2017) outlined the need for semi-supervised feature selections methods in a comprehensive survey. The authors described supervised feature selection methods use labelled data and evaluate specific feature relevance by measuring the correlation between the feature and its class label. Unsupervised feature selection is considered more challenging due to the absence of labels. It evaluates the significance of features by maintaining some of the data characteristics of the data, such as the variance. They continue to explain how, in real-world applications, labelled data availability cannot always be guaranteed and therefore, has created a need for a semi-supervised solution. A semi-supervised feature selection method can use the labelled information where available and the unlabelled data to evaluate feature relevance. The authors concluded that most of the semi-supervised methods surveyed focused on classification problems, and few have experimented with regression problems and recommended this must be further explored.

Qian and Zhai (2013) developed a method that simultaneously performs robust clustering and robust feature selection to select optimal discriminate features for unsupervised learning. The authors demonstrated how unsupervised features selection could be performed by generating cluster indicators using local learning regularised robust non-negative matrix factorisation. Their empirical study on large datasets proved the proposed method was both efficient and scalable. However, unsupervised learning is regarded as the most complex and challenging with the absence of class labels. Qi et al. (2018) take this approach one step further by taking into consideration the correlation between features. The authors demonstrate how introducing an inner product regularisation in their algorithm, which iteratively updates itself efficiently, can achieve better performance than other state-of-the-art unsupervised feature selection methods.

Zeng et al. (2015) introduced a novel feature selection approach by focusing on the feature interactions. The authors argued that traditionally feature selection methods simply focused on removing irrelevant and redundant features and usually ignored the interactions between those features within the data. As a result, fundamentally losing highly relevant features as those only reveal themselves when their relevance is evaluated in combination with other features.

When dealing with regression problems with datasets with many features, there is a greater chance of collinear associations between two or more predictor variables, leading to multicollinearity. Multicollinearity between two or more features can lead to inflating the variance of the regression parameters, errors of the coefficients, diminishing the model's performance (Dormann et al., 2013). By overinflating the variance, significantly relevant feature variables can statistically seem irrelevant (Akinwande et al., 2015). The severity of collinearity can be calculated using the estimated coefficients using the variance-covariance matrix formula (Wonsuk et al., 2013). In general, the aim is to employ a suitable feature selection technique that can reduce the dimensionality of the feature space resulting in a simpler model that is algorithmically efficient and effective while also being able to run on a low powered device (Leppänen et al., 2020).

The reduction of data dimensionality enables the selection of meaningful features and considerably improves learning algorithms' performance (Verónica Bolón-Canedo et al., 2019). However, the main challenge is the lack of a systematic feature selection approach to automatically filter relevant information (K. Chen et al., 2020), mainly because the definition of relevant information is highly problem dependent.

In addition to the lack of a universal approach to select relevant information, feature selection methods still face several challenges (K. Chen et al., 2020). Features sometimes present a structure, such as trees, graphs, overlapping groups, etc. As most algorithms are designed for generic data, they do not assume that features can have correlations. Data that comes from several sources (multisource data) can introduce several different representations of the same feature, which must also be accounted for by selection algorithms. Constantly growing datasets must be stored in other disks/servers (sometimes in different cities or countries). The use of aggregation methods, along with feature selection degrades the scalability of current selection algorithms (Verónica Bolón-Canedo et al., 2019). In worst-case scenarios, only one pass of the data is feasible due to size or complexity, and features must be selected in only one pass. The complexity of datasets leads to data sparsity, which hinders statistically significant results (Z.M. & D.F., 2015). Stability is another concern, such that the algorithm must be able to select the same features even if the original dataset is augmented with noisy data. The difficulty lies in the fact that it is unknown *a priori* if the data belongs to an already clustered feature or if it will introduce a new one. Therefore, a myriad of methods, algorithms and software packages were already proposed to tackle the problem.

The algorithms for feature selection can be structured (Novakovic, 2009):

- Subset generation: a search procedure to generate a subset of features;
- Subset Evaluation: the generated subset must be evaluated by a metric or by a classifier algorithm. If the new subset outperforms the old one (based on a pre-defined metric), it replaces the old subset; Stopping criteria: a threshold to stop the feature selection algorithm. Usually, it halts after a certain number of features were successfully selected, Validation: it compares the efficiency of (machine learning) algorithms between the subset chosen and the original one.

Feature selection methods can also be categorized from the data perspective (J. Li & Liu, 2017), but the categorization above helps subdivide methods into the steps they affect. Before reviewing the feature selection methods, it is worth defining feature relevance. Feature relevance is problem-dependent; therefore, several definitions attempt to characterize "relevant to what?" (Kumar, 2014).

Given a set of features X, a metric J over X is defined such that the selected subset X' of X contains the elements that maximize J in some sense (the subset evaluation). The challenge is to find a compromise between minimizing the cardinality of X' and maximizing J(X'). The search for elements of X to be tested (subset generation) follows the traditional approach, as exponential search, sequential search, random search, evolutionary algorithms etc.

There are other types of subset evaluation methods, commonly based on statistical methods or entropy-based approaches (Novaković et al., 2016). They are also known as rank methods, as they rank each feature of the dataset. Standard statistical methods include chi-squared, one-R and relief-F (V. Bolón-Canedo et al., 2015). Concerning the entropy-based methods, in statistics, entropy measures how unpredictable a system can be. Information gain (IG) defines a metric which compares two features and selects the one with less entropy. Due to a flaw in the IG method (which favours features with more values), the gain ratio (GR) is defined as IG divided by the entropy of the variable being compared. Therefore, the symmetrical uncertainty (SU) is defined as IG divided by the mean of the entropy. IG, GR and SU also provide a measure of the correlation between features.

Statistical methods are also used to evaluate a selected subset. They are also named classification algorithms. Five ways are widely used: instant-based 1 (IB1), Naive Bayes, C4.5 decision tree, artificial neural networks, and radial basis function. IB1 is the simplest nearest neighbour method (Aha et al., 1991). It uses the Euclidean norm to measure the distance between a training feature and a test feature. It is based on the idea that classes similar to the test features are in the neighbourhood in the vector spaces. Naïve Bayes defines discriminant functions based on the Bayes theorem (Maron, 1961). Probabilities and conditional probabilities are evaluated based on the frequency of a feature. It has the advantage to require a small dataset to construct the classifier. C4.5 algorithm fixes one test feature at a time and builds decision trees through comparison with all other features (Quinlan, 2014). It may rely on Naïve-Bayes or IG (entropy-based rank method) to decide each feature's importance. As the algorithm learns, the test features are updated.

The last step in the feature selection process concerns the validation of the selected subset. Criteria are classified with respect to their dependency or independency on algorithms (Kumar, 2014). Those that are independent of a specific algorithm should rely on probability-based measures, like distance, divergence, uncertainty, probability of errors, consistency and interclass distance. The dependency on algorithms is indirectly measured through improved performance, generalization capacity, or time-to-solution.

Unsupervised feature selection (UFS) are unbiased and reduce overfitting (Devakumari & Thangavel, 2010). They can be split into three categories: filter (properties of the data are used to evaluate features, without using clustering algorithm), wrapper (attributes are assessed based on the results of a clustering algorithm) and hybrid. They are moving from the filter to wrapper methods shifts from computational scalability and efficiency to robustness. UFS methods use three approaches to select features: (i) it determines the features that can more effectively preserve the manifold structure of the given dataset; (ii) it searches for clusters of data, in which unsupervised methods become supervised methods; (iii) it selects features with higher or lower correlation. The latter also determines features redundancy. Solorio-Fernández et al. (2020) provide an excellent review and comparison of most referenced methods.

Of particular interest are the supervised feature selection methods, in which algorithms learn from available labelled data and the abundant unlabelled data (Sheikhpour et al., 2017). The methods can be categorized into generative, self-training, co-training, semi-supervised support vector machines, and graph-based processes. Again, the split between filter, wrapper and embedded methods holds. Sheikhpour et al., 2017 provides a broad overview and taxonomy of the methods available.

Table 1 lists the summarised most common filter, wrapper and embedded methods. For details, see the comprehensible reviews of feature selection methods and references therein (Wang et al., 2014; Bolón-Canedo et al., 2014; Jović et al., 2015; Hira & Gillies, 2015; Bolón-Canedo et al., 2015; Sheikhpour et al., 2017; Colaco et al., 2019; Remeseiro et al., 2019) and the book (Bolón-Canedo et al., 2015).

Table 1. Summary of feature selection approaches.

Method	Taxonomy	Reference
Monotone Dependency	Filter	Bolón-Canedo et al., 2011
Multitask Filter (M_FS)		Lan et al., 2011
MASSIVE		Meyer et al., 2008
Maximum Weight Minimum Redundancy		Wang et al., 2013
Partial Least Squares		Student et al., 2012
Redundancy Feature Selection		Ferreira et al., 2012
Relevance Redundancy Feature Selection		Nie et al., 2010
t-test Feature Selection		Jafari et al., 2006

Bayesian Networks		Hruschka et al., 2004
Gain ratio		Witten & Frank, 2011
Symmetrical Uncertainty		Yu & Liu, 2003
Correlation		Yu & Liu, 2003
Chi-square		Witten & Frank, 2011
Inconsistency Criterion		Liu & Setiono, 1996
Fisher Score		Duda et al., 2006
Spectral Feature Selection (SPEC) and Laplacian Score (LS)		Alelyani et al., 2013
Feature Selection for Sparce Clustering		Witten & Tibshirani, 2010
Localized Feature Selection based on Scatter Separability (LFSBSS)		Li, Dong, Hua, 2008
Multi-Cluster Feature Selection (MCFS)		Alelyani et al., 2013
Feature Weighting K-means		Modha & Spangler, 2003
ReliefC		Dash & Ong, 2011
Supervised – Deep Neural Network		Cheng and Scotland, 2017
Supervised – Transfer Learning		Akbari et al., 2019
Supervised – Active Learning		Gudur et al., 2019
Supervised – Attention Mechanisms		Vishvak et al., 2018
Supervised – Convolutional Neural Networks		Hammerla et al., 2016
Supervised – Recurrent Neural Networks		Hammerla et al., 2016
Supervised – Deep Belief Network		Zhang et al., 2015c
Supervised – Restricted Boltzmann Machine		Radu et al., 2016
Supervised – Stacked Autoencoder		Almaslukh et al., 2017
Genetic Algorithms (GA)	Wrapper	Jirapech-Umpai et al., 2005
GA-KDE-Bayes		Wanderley et al., 2013
Successive Feature Selection		Sharma et al., 2012
Sequential Search		Glass et al., 1965
FRFS	Embedded	Wang et al., 2013

Iterative Perturbation Meth. (IFP)		Canul-Reich, et al. 2012
Kernel Penalized SVM		Maldonado et al., 2011
Probably Approximately Correct (PAC) Bayes		Shah et al., 2012
Random forest		Anaissi et al., 2011
SVM Recursive Feature Elimination		Guyon et al., 2002
Least Absolute Shrinkage and Selection Operator (LASSO)		Bellal et al., 2012

Feature selection is indispensable in human PA recognition studies, as this classification problem relies on video-based or sensor-based data, which must be pre-processed before analysed. Video-based data demands higher computational power to be evaluated and raises issues about privacy disclosure. Although sensor-based data are very time-consuming to collect and annotate, they are easier to anonymize, but still pose numerous computational challenges to feature selection (K. Chen et al., 2020). Many activities feature intra-activity similarities. For example, walking and running are different activities with very similar characteristics, that can be performed differently by distinct individuals. The data contains traces of personality, and it is, therefore, challenging to be uniquely associated with an activity. The collected data will likely be imbalanced because it will not contain information from all possible scenarios (e.g., domestic accidents). This compromises the generalization of machine learning algorithms. Also, humans typically perform activities concurrently, which implies that the data from sensors may represent a composite of activities, including the presence of several individuals. Therefore, the accuracy of PA recognition heavily depends on precise feature extraction. The process of feature selections can be of three folds as described by Guyon & Elisseeff (2003); firstly, to improve predictors' performance, secondly, finding cost-effective and faster predictors. Finally, it seeks to understand better which predictor variables selected can reduce the computational load on low powered devices.

2.1.2 Lightweight ML

Agarwal & Alam, (2020) developed a lightweight machine model using a shallow Recurrent Neural Network (RNN) combined with a Long Short Term Memory (LSTM) deep learning algorithm to produce an effective model that could be deployed on edge devices. The authors demonstrated their model's usefulness for Human Activity Recognition (HAR) which requires less computational power and is efficient enough to deploy on edge devices. The team used an Android smartphone to capture accelerometer readings to determine activities that included walking, climbing stairs and up and down, standing and sitting. Their experiment, which contained 29 subjects showed it was possible to achieve above 81% accuracy in classifying the given activity.

Tang et al. (2020) presented a novel lego-CNN approach that reduces memory and computational cost while maintaining comparable accuracy. The authors explained the limitations of the conventional use of Convolutional Neural Networks (CNN) for HAR data modelling. They explained that as it requires more computing resources, it limits its application on edge devices. The authors also addressed that CNNs will often need special network architectures for visual tasks which are not suitable for HAR tasks with time-series signals. The authors proposed using a set of lower-dimensional filters as lego bricks to be stacked for conventional filters that do not rely on a special network structure. Zhang et al. (2019) proposed a systematic approach to developing a lightweight HAR algorithm using a one-dimensional CNN and applied a Singular Value Decomposition to compress the weights in the fully connected layers of the 1D-CNN. The authors demonstrated the effective use of their lightweight ML approach that leverages a low-complexity 1D-CNN for real-time skeleton-based HAR.

Punithavathi et al. (2019) presented the need for a highly secure authentication mechanism and proposed a framework for the lightweight cancelable biometric authentication system. Their objective was to deploy this approach in real-world settings which offered a high level of accuracy and had minimal overhead by consuming less time to authenticate.

Bikmukhamedov & Nadeev (2019) introduced a new flow feature of a categorical type that describes a set of Transmission Control Protocol flag fields within a flow. The authors employed the Principle Component Analysis (PCA) method to remove correlated features and perform feature transformation. Their approach resulted in an F1-measure for logistic regression from 99.1% up to 99.6%, which was comparable to computationally expensive models and demonstrated the feasibility of their lightweight model for IoT based applications. H. Lu et al. (2020) introduced Deep Reinforcement Learning (DRL) to solve the offloading problem using mobile edge computing and an LSTM network.

Lee et al. (2020) designed and developed a lightweight ML-based intrusion detection system specifically for resource-constrained devices. The authors demonstrated the effective use of their lightweight ML model using a stacked encoder, mutual information and C4.8 wrapper. Using the Aegean Wi-Fi Intrusion Dataset (AWID), their model achieved above 98% accuracy in identifying a genuine intrusion which is on par with existing research. Sakr et al. (2020) introduced the Edge Learning Machine (ELM), an ML framework for edge devices which performs inferences on microcontrollers. The authors successfully deployed several different ML models on STM32 Nucleo boards. Their results showed that it was possible to achieve the same score as on a desktop computer in their experiments. However, several factors impacted the performance and found that in some cases, there was a significant memory demand when the input data size increased, rendering the model less efficient on the edge device.

2.1.3 Pipeline Optimisation for an Optimal Model

ML problems can be solved using many different approaches, and the choice of which algorithm that should be used depends on several factors. Factors such as the type of dataset, the structure of the dataset, the dataset's size, and the analytical objectives play a large part in the decision-making process. More importantly, while evaluating the algorithms, the practitioner's judgment plays a large role in achieving optimal performance (Elkan, 2010). J. Chen & Ran (2019) explained as deep learning is rapidly evolving researchers are finding that choosing the appropriate algorithm is challenging. The authors described this to the lack of apples-to-apples comparison to the target device. Several steps within the ML pipeline need to be considered before successfully deploying on the edge device. The typical ML pipeline usually consists of the following three core steps; data preparation, data modelling and model evaluation (Kotu & Deshpande, 2015; Lechevalier et al., 2015). However, as this task increasingly becomes more complex and labour intensive, there is a growing shift towards the use and the need for automated optimisation techniques. Sensor readings can quickly produce large volumes of data, and some may need to be discarded to maintain additional computation resources for more critical tasks. In this section, some of the recent studies in suitable model selection and the automatic optimal ML model selection techniques are outlined that may contribute to deploying on edge devices effectively.

With so many ML models to choose from, selecting the most effective one is a challenge of its own. In addition to this, tuning the various parameters can be just as labour intensive and time-consuming. The concept of automated ML aims to solve this by enabling practitioners to quickly evaluate all the possibilities and narrow down the selection process while also performing feature pre-processing and hyperparameter optimisation (Feurer et al., 2015a).

Traditionally, research in automated ML focused on the automation and optimisation of the ML process's subtasks. As the various studies have matured, it has given rise to efforts to develop a fully automated machine learning pipeline. Feurer et al., (2015b) motivated by AUTO-WEKA developed AutoML that employed the *Combined Algorithm Selection and Hyperparameter* (CASH) optimisation technique to automate the ML pipeline. The AutoML system initially relied primarily on an efficient Bayesian optimisation technique for hyperparameter tuning, and its effectiveness was evaluated in various classification problems. The team have since further improved their system and introduced a Robust new AutoML system based on Scikit-learn that automatically improves by taking into account past the performance of similar datasets (Feurer et al., 2015a). The team have named their new system Auto-sklearn, which has outperformed many other AutoML systems at the ChaLearn AutoML challenge 2015 (Guyon et al., 2015). However, Auto-sklearn was not designed to perform well with large datasets and secondly, not yet fully developed to deal with regression problems (Hu & Huang, 2017; Olson et al., 2016). Klein et al., (2017) have developed RoBO, a system that implements Multi-task Bayesian Optimisation to find a global optimiser out of the black box function through noisy observations and Fabolas to speed up hyperparameter optimisation which is particularly useful on large datasets. These advances are ideal for improving the computational efficiency of the ML model for deploying on edge devices.

Olson et al. (2016) developed a Tree-based Pipeline Optimisation Tool (TPOT) to automate the ML pipeline and demonstrated its effectiveness in classification performance using a series of real-world datasets. By adopting genetic programming techniques to evolve pipeline operators and their parameters, the team were able to maximise the classification accuracy. Secondly, by integrating the Pareto optimisation technique into their system, the authors could also demonstrate its ability to minimise the pipeline's complexity.

Based on the evidence thus far, automated ML has many benefits and can aid in producing significant performance gains over traditional methods, especially in creating good baselines. However, most of the research focuses on classification problems and fails to address a broader range of machine learning tasks such as automating the regression problems workflow. Secondly, there is little evidence of automated approaches to solving the problem of feature multicollinearity that negatively impacts the predictor performance. Given the volume of research focused on automating ML pipelines, it is now possible for us to explore how we can harness the key strengths from each. Create a balanced, holistic approach to optimise the automated feature selection process further and improve the effectiveness and computational efficiency of the auto ML workflow for deploying on edge devices.

Fergus et al., (2017) suggested the following well-known models are commonly employed for evaluating and comparing performance: Linear Regression (LR), Ridge Regression (RR), Multi-Layer Perceptron (MLP), Random Forests (RF), Convolutional Neural Network (CNN), gradient boosted decision trees, XGBoost (XGB), Support vector machine (SVM), Decision tree (DT), Naïve Bayes (NB) and k-Nearest Neighbour (NN). However, others have also indicated Logistic Regression and Linear Discriminant Analysis are also suitable candidates (Christofaro et al., 2018; Hassan et al., 2018) for deploying and evaluating the performance on edge devices.

2.1.4 Predicting Energy Expenditure in Physical Activity

The use of accelerometry data produced by wearable devices has been increasing over the past ten years to estimate Physical Activity (PA) in children (Crouter et al., 2018; Alex V. Rowlands et al., 2013). Despite this, the accuracy of accelerometer derived PA data compared to actual energy expenditure is linked to the age group of children, the model of accelerometer and wear location. Although some studies have examined the utility of machine learning approaches to predict accelerometer derived PA (Montoye, Moore, et al., 2018; Montoye, Westgate, et al., 2018), few of these have examined children precisely and none, to date, have included activities representative of children's fundamental movement skills. This is a key, but the under-examined issue as children's PA tends to be sporadic and omnidirectional in nature (Rowlands & Eston, 2007). Holfelder & Schott (2014), explained that accelerometer using locomotor activities derived accelerometer cut-points may not accurately reflect the actual physical activity levels of children. Research has suggested that by explicitly understanding how the various types of repetitive object control skills contribute to activity intensity (Sacko et al., 2018). And, there are very few studies that have examined accelerometer performance compared to energy expenditure (MET values) associated with object control skills in children (Sacko et al., 2018). Others, such as Blythe et al., (2017) & Chowdhury et al., (2017), have found that whilst innovation in wearable technology has improved in accuracy using consumer-grade devices, they have failed to perform at an equivalent level with research-grade equipment. They have recommended further research, and an independent quality standard of accuracy rates be explored.

Recent improvements in wearable devices, such as high-frequency data sampling and advances in ML, have enabled the potential for accelerometers in PA monitoring. Despite this, the accuracy of accelerometer derived PA data compared to actual energy expenditure is still specific to the age group of children and the physical wear location.

There is a need to improve the accuracy of accelerometry derived analysis to understand better PA's influence on children's health (Crouter et al., 2018). The majority of studies using machine learning techniques have, to date, only examined wrist and/or waist located accelerometers. Observations of misclassifying in cycling activities have been found in adult-based studies (Manini et al., 2014) where the wrist position during cycling may result in activity intensity being misclassified when using wrist-worn sensors. Ankle worn accelerometer-based wearables may be a more suitable option that could better reflect cycling activities. One recent study (Duncan et al., 2019) has identified that ankle worn accelerometry may be better than waist or wrist-worn accelerometers in assessing moderate-intensity PA in children. Additional work is needed to support the assertions of Duncan et al. (2019). To date, few studies have examined the utility of machine learning approaches in classifying PA in children which includes fundamental movement skills and comprises accelerometers worn at several locations.

Measuring the metabolic equivalent (MET) provides a convenient method to determine an individual's tolerances during physical activity in which a person may safely engage in without exceeding their prescribed intensity levels (Blumchent, 1990). Wearable devices are packed with many sensors and usually include one or more of the following, accelerometer, gyroscope and or pressure sensors. As a result, they are becoming more accessible due to their small size, portability, low power consumption and low cost (Alinia et al., 2015). With the increasing popularity of wearable technology and advances in machine learning creates the ideal opportunities to explore and evaluate potential approaches to accurately measure MET values outside of the laboratory using low powered devices (Yavelberg et al., 2018).

Traditionally, physical activity-based monitors using the accelerometer readings converted the raw data into activity counts, applications commonly known as pedometers, which was matched to frequency and magnitude of acceleration (H.K. Montoye et al., 2016). Thresholds were developed and called 'cut-point' which were used to evaluate physical activity intensities from the accelerometer data (H.K. Montoye et al., 2016). However, this method proved inadequate in accurately determining physical activity intensities and failed to differentiate between standing, sitting, and or lying down positions (H.K. Montoye et al., 2016). The researcher has since looked at machine learning approaches to help improve the accuracy of the METs of physical activity measurements (H.K. Montoye et al., 2016). The studies have shown that machine learning models have drastically improved the MET measurement accuracies of physical activity using data generated on accelerometer-based wearable devices (H.K. Montoye et al., 2016; Montoye et al., 2015; Preece et al., 2009).

The MET values derived from the energy expenditure in children and adolescents are significantly lower than adult (Lyden et al., 2013). However, there is little or no evidence of similar studies on children that accurately measure METs of physical activity outside lab conditions using modern machine learning models and data from low powered wearables devices.

2.1.5 Swim Activity Recognition

Swimming is one of the few sports that people of all ages and abilities can take part in, that involves the rhythmic movement of many muscle groups for an all-over workout (Mooney et al., 2017). Recent advances in wearable technologies have paved the way for better tracking of human Physical Activity (PA) in general. Moreover, the advent of waterproof wearable devices, such as the Samsung Gear Fit, Apple Watches, Fitbit and Misfit Shine, has caused an increase in pervasive PA tracking in swimming (Kos & Umek, 2019; Mooney et al., 2015), with significant advantages over traditional computer vision-based approaches (Mooney et al., 2017), which were already used to detect possible drowning incidents (W. Lu & Tan, 2002). The advances in kinematic swim sensor technology have allowed researchers to analyse physical stroke mechanics, swim classification (Fani et al., 2018; Omae et al., 2017; Siirtola et al., 2011) and evaluation of exercise intensities (Mooney et al., 2015). By monitoring PA, world-class swimming athletes benefit from being able to measure their performance over time during their preparation for competition (Mooney et al., 2015). Practitioners can also measure their performance as a motivational tool (Mooney et al., 2017) and therapists can provide real-time feedback to patients who need swimming exercise for physical rehabilitation (Kos & Umek, 2019).

Researchers carried out many experiments to classify swimming styles using inertial sensors focused on rough detection of swimming motions to help determine the swimming style (Hakozaki et al., 2018; Omae et al., 2020). Omae et al, took their original approach a step further to detect the exact timings of the swimming motion. The device which was 67mm x 26mm in dimension was attached to the lower back of the swimmer's waist. The authors constructed an algorithm which extracts the sub-data from the sensors and using a window length of half the average time of the butterfly stroke was able to pre-process the input data using the Fast Fourier transform (FFT) to feed a CNN model. The CNN model was able to detect the start timings utilising the sensor data from the second half of the stroke. Their model consisted of two convolutional layers and three fully connected layers in the middle. Their algorithm was able to achieve a precision and recall of 0.855 and 0.904 respectively and proved their algorithm was useful in classifying the butterfly swim style. In their earlier research, the authors introduced ensemble learning and demonstrated that it was possible to improve the mean value of the F-measure for breaststroke and butterfly and were able to achieve 0.981 mean F-measure (Omae et al., 2017). In a separate experiment, the team were able to develop a swim motion estimation method for turn detection using random forest classifier (Kobayashi et al., 2019).

Another team of researchers were able to distinguish the efficient high elbow pose using the sensor data to help swimmers improve their freestyle (crawl) swimming motion (Fani et al., 2018). The authors employed a random forest classifier and were able to yield a 67% accuracy in determining the ideal most efficient pulling stroke. A separate group of researchers, (Marinho et al., 2020), looked at the start, turn and finish performances who found that the underwater variables were the main contributing factors to a faster start and an efficient swimmer; their analysis can be used for improving the swimmer's performance. Polach et al., 2019, found that swimming performance can be divided into several essential parts, start, swim stroke, turns and finish. The authors found that the turns played a significant role in improving the swim time. The authors analysed 59 turns and found that they played a significant role in the final swim performance and better analysing the turns can help swimmers improve their performance. Specialised swimming activity monitors are also found to be inaccurately measuring lap timings in professional settings, especially in the first and last laps of the swimming sessions (Mooney et al., 2017). These miss-timings caused erroneous calculations in the swim metrics such as stroke counts/rates. The authors concluded that with further research and development of the feature detection algorithms can help to increase the accuracy and increase the suitability in competitive settings.

2.1.6 Activity Recognition in Smart Homes

Increasing research in activity recognition (AR) has become possible with the growing availability of low-powered wireless sensor networks that can continuously stream ambient data and enable researchers to draw inferences from (Mark Eastwood et al., 2019; Konios et al., 2018). By better understanding, the resident's normal activity patterns such as making breakfast, eating lunch and sleeping etc. provide a reasonable indication that the resident is doing well. This also allows the system to recognise patterns in the resident's activities that fall outside normal behaviour such as a fall and indicate a problem and trigger an alert for help (Eastwood et al., 2019). A useful AR model is required for such a system to operate at a reasonable standard that doesn't trigger unnecessary alerts and ensures that genuine causes for concern are raised immediately.

As people age, their various needs are continually changing, and the number of people ageing population is increasing steadily (Helal & Bull, 2019). The smart home is a residential setting that is equipped with a variety of IoT based sensors that monitor the condition of the living space and its occupants (M Eastwood et al., 2019; M. Li et al., 2018).

Liciotti et al. (2019), provided a useful overview of the various classification algorithms used commonly for AR, such as Naïve Bayes (NB), Random Forests (RF), Hidden Markov model (HMM), Conditional Random Fields (CRF), k-nearest neighbour (k-NN) and Support Vector Machines (SVM). And, explained that these algorithms produced static models that did not adapt to the changing environment as they were less discriminative. Instead, the authors proposed a sequential deep learning LSTM approach to HAR and described its advantages over traditional models. The authors pointed out that LSTM allows automatic learning of Spatio-temporal information from the sensor reading without the need for handcrafting features and secondly, it can model the temporal evolutions of the features using recurrent connections within the hidden layers. Their experiments using LSTM approach and additional variations of the model (Uni-LSTM, Bi-LSTM, Casc-LSTM and various combinations) showed that their approach does perform significantly better in HAR in a smart home setting over traditional ML approaches using five different CASAS datasets. Gil-Martín et al. (2020) explained that most previous studies divided the acceleration signals into overlapping windows and provided the recognised activity per window. The authors pointed out that AR often continues and lasts longer than the given window or overlapping window and thus would be necessary to extend this period or group the windows over a more extended period and then attempt to classify the activity to enhance the model's accuracy. The authors proposed a new approach which consists of three main components. Their first component segments the accelerometer readings into overlapping windows extracting the frequency information using Fast Fourier Transformation (FFT) and focusing on the low frequencies. The authors explained the use of FFT was particularly useful in capturing discriminate features from sedentary to vigorous activities. The second detects the movement performed within each window using a deep learning model based on CNN, and the third component integrates the window level decision for more extended periods to gain a significant improvement in performance.

Conditional Random Fields (CRFs) are discriminative, as a finite state model, that uses a single exponential model for the joint probability distribution of the entire sequence (Lafferty et al., 2001). CFRs are commonly used in sequence labelling tasks as the model doesn't rely on dependencies and as a result, avoids label bias (Eastwood et al., 2019; Lafferty et al., 2001). Abidine et al., (2018) described how the joint use of sequence features combinations combined with a modified weighted SVM can achieve performance improvements and are more efficient over traditional sequence analysis methods in daily AR. However, found that the predictions were vulnerable to class-overlap and the location of the sensors influenced the recognition performance.

Based on the work carried out by Eastwood et al. (2019), AR was considered a sequential prediction problem. It was assumed that the sequence of observations $X =< x_{t1}\ x_{t2}\ \dots\ x_{tn} >$ each labelled with the action that was being performed Y, where each x_t belongs to a set of possible sensor readings X, and each label y_t to a set of possible labels Y. Their task was to predict the correct labels of an unknown sequence of observational readings. A Conditional Random Field (CRF) is a model used for predicting the most likely sequence of labels that correspond to a sequence of observational inputs that are of similar form (Lafferty et al., 2001). It is also understood that the model's performance is directly affected by the features used in training the model (Eastwood et al., 2019).

Typically, generating features requires extrapolating features that follow a specific pattern or template and to create complex features there is a need for predefined complex templates (Eastwood et al., 2019; Sutton, 2012). However, this can result in generating many features that can adversely affect the performance of the model. Eastwood et al. (2019) introduced an approach to extract features using decision trees to select a small sub-set of complex features that increase the model's predictive performance.

Ambient assisted living (AAL) is primarily focused on helping people in their natural environment. The stakeholders of ALL are not limited to the resident but also include social services, health workers, relatives and other care agencies (Calvaresi et al., 2017). By collecting information generated by the various sensors in the smart home, it's possible to improve the quality of life, which is at the core of the AAL (Acampora et al., 2013). Smart home environments have become far more sophisticated by making use of modern technologies such as embed sensors in smaller more widely available microcontrollers and devices that utilise Bluetooth Low Energy (BLE), Zigbee and RFID for wireless communication (Eastwood et al., 2019). Smart home sensors can range from non-invasive infra-red motion sensors to various environmental sensors such as humidity, temperature, door (open or close), motion detection and power usage sensors. By leveraging machine learning's predictive capabilities, it's possible to gather the various ambient data streams to a central remote server and analyse the data for AR in supporting and enhancing AAL.

Human activity-based datasets usually consist of a mixture of regular (sequential) and irregular events of varying durations. In other words, human activity based sequential data contains time misaligned events (Narimatsu & Kasai, 2015). Therefore, there is a need for a *naive* approach that can extrapolate patterns of varying duration and sequence. The Hidden Markov Model (HMM) statistical modelling technique has been widely used for use in sequence modelling using transition probability. HMM models employ a finite set of states, each associated with a multidimensional probability distribution (Dahmen et al., 2017). The transitions between several hidden states correspond to specific sequence label (Eastwood et al., 2019). The associated distribution of output allows inference of the probability state given a sequence of observations (Eastwood et al., 2019). There are many various of HMM such as Hierarchical HHMM described by Fine et al. (1998), as suitable for learning sequential patterns over multiple scales. Semi-Markov SHMM and CRFs methods that are modified to better model states focusing on durations to map observed sensor readings to the hidden activity states (Van Kasteren et al., 2010). Conditional Random Fields (CRFs) are discriminative, as a finite state model, that uses a single exponential model for the joint probability distribution of the entire sequence (Lafferty et al., 2001). CFRs are commonly used in sequence labelling tasks as the model doesn't rely on dependencies and as a result, avoids label bias (Eastwood et al., 2019; Lafferty et al., 2001). The authors described how a Support Vector Machine (SVM) transform the data that needs to be classified into a high-dimensional space by using the Kernel trick.

In which the separating hyperplane is learnt by using maximum-margin. Guo et al., (2018) have introduced a Support Vector Machine (SVM) based effective sequential classifier for multitemporal remove image classification. Guo et al. (2018) proposed an approach that leveraged the temporal trend of a set of previous classifications and fine-tuned into more accurate positions of sequential images. Guo et al. (2018) described how a binary linear SVM could be extended to handle a multi-class case by applying the *one-against-one* strategy. However, the authors found that the performance of their proposed classifier was significantly depended on the previous classifiers' accuracies and that this problem was more critical when there were few training samples. Cook et al., (2013) found that SVMs consistently performed better than other approaches and described how a fixed dimensional feature vector could be used to train an SVM model using the time span of the k-event sensor window, the time of day and the number of events in the sensor window. Abidine et al., (2018) described how the joint use of sequence features combinations combined with a modified weighted SVM can achieve performance improvements and are more efficient over traditional sequence analysis methods in daily AR. However, found that the predictions were vulnerable to class-overlap and the location of the sensors influenced the recognition performance.

Most senior citizens care deeply about maintaining as much of their independence as possible, however still want help to be quickly available when required, which poses a problem for care home settings (Eastwood et al., 2019). The technology can also enable relatives to monitor their loved ones. Smart home wireless activity recognition technology paired with data analytics can provide the extra level of assurance the social care system needs while preserving their patient's independence (Eastwood et al., 2019). A device that utilises a well-being indicator that sounds an alarm in case of abnormal behaviour such as an accident or health problem relies on AR to trigger the alarm after identifying inconstancies in the resident's activity (Jing et al., 2017).

2.2 Summary

The following are conclusions of the research challenges and gaps within the current related work and which have given rise to some research question.

Based on the current related works, lightweight ML has many benefits and can clearly aid in producing significant performance gains over traditional methods, especially in setting good baselines. However, most of the research focuses on classification problems and fails to address a broader range of ML tasks such as automating the regression and classification problems. Secondly, there is little evidence of automated approaches to solving the problem of feature multicollinearity that negatively impacts the predictor performance. Given the volume of research focused on automating ML pipelines, it is now possible for us to explore how we can harness the key strengths from each and create a balanced, holistic approach to optimise the automated feature selection process further and improve the effectiveness and efficiency of the automated ML workflow. This gives rise to the following question: **Q1: Is it possible to automate feature selection and optimisation in order to improve the performance of a lightweight ML model?**

As children's PA is omnidirectional in nature (Rowlands & Eston, 2007), despite this, the accuracy of accelerometer derived PA data compared to actual energy expenditure is still specific to the age group of children and the physical wear location. Although some studies have examined the utility of machine learning approaches to predict accelerometer derived PA (Montoye, Moore, et al., 2018; Montoye, Westgate, et al., 2018), few of these have examined children precisely and none, to date, have included activities representative of children's fundament movement skills. Others, such as Blythe et al., (2017) & Chowdhury et al., (2017), have found that whilst innovation in wearable technology has improved in accuracy using consumer-grade devices, they have failed to perform at an equivalent level with research-grade equipment. Based on these findings: **Q2. Is it possible to apply a lightweight ML approach to effectively predict the Metabolic Equivalents (Energy Expenditure) in children using low powered devices?**

The advances in kinematic swim sensor technology have allowed researchers to analyse physical stroke mechanics, swim performance and evaluation of exercise intensities (Mooney et al., 2015).

The authors also indicate that technology also offers significant advantages over traditional computer vision-based approaches. Mooney et al., (2017) have also found that lap timings were not accurately timed and were especially the case in the swimming sessions' first and last laps. These miss-timings also caused the swim metrics such as stroke counts/rates to be miss-calculated as a direct result. The authors concluded that with further research and development of the feature detection algorithms, it could increase accuracy and increase competitive settings' suitability. This has given rise to the final question: **Q3. Is it possible to apply a lightweight ML approach to classify swim activities using low powered devices accurately?**

Chapter 3: Methodology

3.1. The Novel Feature Selection Approach for Lightweight ML

Formulating a problem definition is usually the first step to initiate a typical ML workflow, and it is generally outlined by selecting a feature from within the given dataset to model and predict against as the target value (Kanter & Veeramachaneni, 2015). There are three critical steps in this approach, and they are feature pre-processing (in green), feature selection (in blue) and the model selection, illustrated in ***Figure 4***. The proposed approach's significant contribution focuses on the feature selection steps, highlighted in blue in ***Figure 4***. The new feature selections approach was developed to achieve significantly greater computational efficiency towards ML models' deployment on low-powered devices.

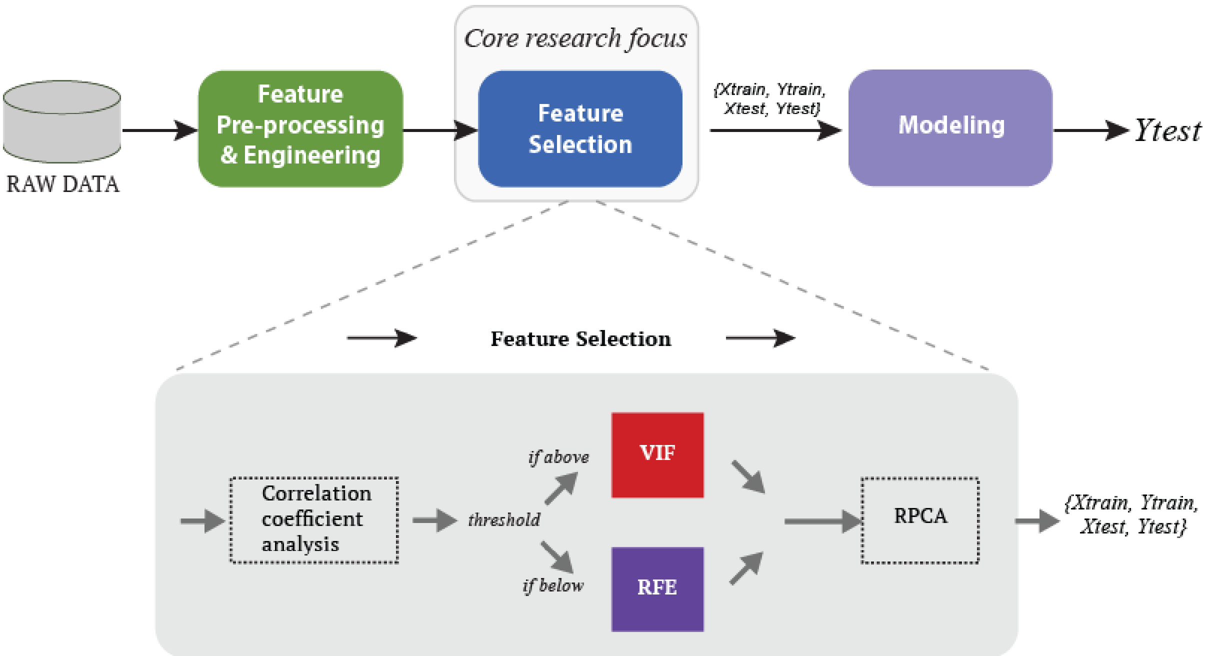


Figure 4. The novel feature selection approach towards lightweight ML

The purpose of feature prepossessing is to enhance the associated components that are relevant to the given problem to be carefully considered. The process of feature prepossessing and feature engineering steps are illustrated in the green box in ***Figure 4***. Feature engineering is traditionally an iterative process, which requires manual intervention and involves several tasks such as removing bad data, removing missing values, transforming and or aggregating values. In this approach, the pre-possessing step will use the Scikit-learn library to sanitise the data, run appropriate imputation methods to deal with missing values. This iterative process built into auto-sklearn can automatically evaluate features against the performance of a simple linear model, such as linear or ridge regression until an acceptable accuracy is achieved. The Multiple imputation method will be used for classification problems and Mean values used for regression problems. Secondly, perform any necessary categorical data encoding such as one-hot encoding and normalise the features. Finally, run the elements through the Feature Tools library, to perform feature engineering by stacking multiple transformations and aggregations operations on feature primitives to create new potentially relevant features. Feature engineering can drastically improve the performance of ML algorithms, however, can increase the complexity of the data model, reducing its efficiency, hence the need for rigorous feature selection approach.

In order to run ML models efficiently on an edge device, a novel automatic feature selection approach has been developed, shown in ***Figure 5***. The novelty of the proposed approach is based on the combination of Variance Inflation Factor (VIF), Recursive Feature Elimination (RFE) and Randomised version of the Principle Component Analysis (RPCA) for feature reduction. The first step of reduction is based either on VIF or RFE, and the choice depends on the values of the correlation matrix (high correlation → VIF, low correlation → RFE). VIF is computationally more efficient than RFE, because RFE includes a built-in modelling algorithm that is run iteratively whereas VIF is a single pass filter reduction. In case of low correlation between features, VIF detection of multicollinearity between features is not performed, and the algorithm takes the RFE branches. In the case of low feature correlation, the RFE algorithm can achieve acceptable results and reduce the number of features while it is computationally more demanding than VIF; however, less computationally demanding than RPCA alone. The approach's objective is to reduce the number of features for improving data modelling efficiency while also being effective on low powered devices. Using RFE does not impact the data modelling efficiency but instead improves it by feeding it a reduced feature subset. Furthermore the proposed new algorithm to obtain a reduced model using a reduced feature subset, is considered lightweight because it identifies a case where the usage of RFE can be avoided (high correlation between some features) in favour of a more efficient VIF. However, the synergy between the two algorithms (VIF, RFE) is necessary to obtain a reduced model in both cases of high and low correlated features.

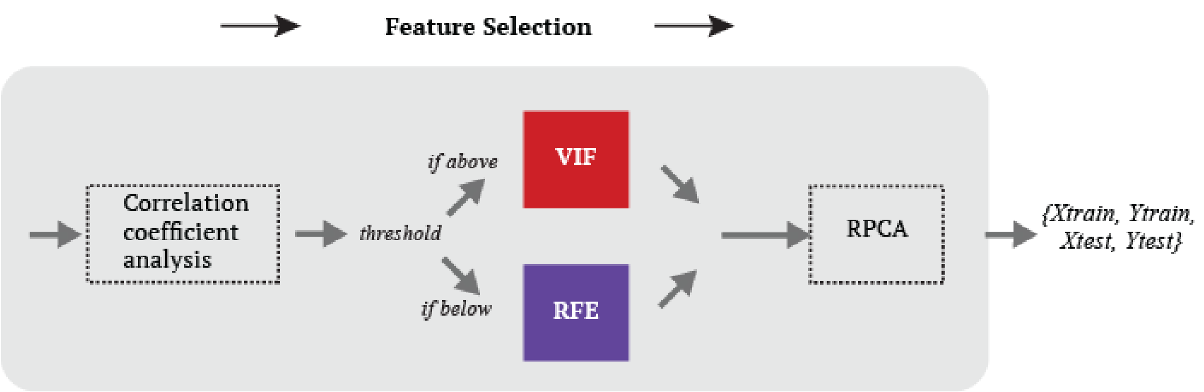


Figure 5. The Feature Selection step breakdown

Equation 1 refers to the correlation coefficient analysis adapted from the Pearson Correlation Coefficient formula. The pairwise coefficient C_{ij} represents the correlation coefficient between feature X_i and X_j. The value of C_{ij} is used as a measure that is evaluated against the specified threshold C_{max}. Each feature is a numerical array (time series) of length *N*, and the summations are extended from 1 to *N* index of the given features.

$$c_{ij} = \frac{\sum_n (x_{i,n} - \bar{x}_i)(x_{j,n} - \bar{x}_j)}{\sqrt{\sum_n (x_{i,n} - \bar{x}_i)^2}\sqrt{\sum_n (x_{j,n} - \bar{x}_j)^2}} \tag{1}$$

To decide whether to apply the VIF or the RFE method, one should determine whether there are “any” features correlated or “none”. VIF is likely to give information useful for feature elimination when there exist elements that are correlated. In contrast, RFE does not require that and will work effectively with uncorrelated features. Next, compute the correlation matrix and look for the elements that are greater than a threshold. If any exist, then the algorithm will perform the VIF based reduction; otherwise, it will show the RFE based reduction.

If

$$c_{ij} \leq c_{max} \quad (2)$$

For each, i and j then perform the RFE feature elimination. Otherwise, if there exists at least an i index and a j index such that (3), then complete the VIF based feature elimination.

$$c_{ij} > c_{max} \quad (3)$$

Note that if two features are $C_{ij} > C_{max}$ then they are considered correlated, such that when they correlate higher than C_{max} = 0.8. If one performs correlation-based feature reduction, then features with a correlation higher than 0.8 can be removed using VIF and if none are correlated RFE can be used. Recursive feature elimination involves the identification of features that underperform within its internal modelling process thus removing them one by one. The elements of correlation matrix C_{ij} are checked one by one against the specified threshold C_{max}. Therefore RFE should only be used when the outcome of the analysis to determine which features are not correlated and are all lower than C_{max}. Combining the information provided by multiple statistical analyses, it's possible to enhance the accuracy associated with the selection process of the essential elements.

Variance Inflation Factor estimator is introduced to the approach. To define VIF, consider the linear regression model with k independent variables (4). It can be proved that the assessed variance of the coefficient estimation β_j, can be expressed similarly as in equation (5), where s is the root mean squared error, X is the regression design matrix ($x_{i,j}$ is the value of the independent variable j-th for the i-th observation), R_j^2 is the coefficient of evaluated regression of X_j on the remaining covariates X_k (with k other than j).

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k + \epsilon \quad (4)$$

$$\widehat{\mathrm{var}}(\hat{\beta}_j) = \frac{s^2}{(n-1)\widehat{\mathrm{var}}(X_j)} \cdot \frac{1}{1 - R_j^2} \quad (5)$$

The VIF is defined as in **(6)** and equals 1 when the vector X_j is statistically independent to each column of the matrix for the regression of X_j on the remaining covariates. In general, the vector X_j is not statistically independent to all columns of the given matrix for the regression of X_j on the remaining covariates, and the value of VIF becomes higher than 1 and represents the magnitude of multicollinearity between the regressor variables.

$$VIF_j = \frac{1}{1 - R_j^2} \tag{6}$$

The Variance Inflation Factor (VIF) provides an additional indicator for feature elimination. Typically, those features that have a variance threshold greater than 5 can be eliminated because of their collinearity with other features.

The Recursive Feature Elimination strategy (RFE) (7) is also used to remove features that do not meet the minimum variance threshold. RFE strategy recursively eliminates features by building a model, in this case, randomised decision trees, on the features that remain and ranks the features based on their predictive score against the target Y values.

$$X_{red,RFE} = \Psi_{red,RFE}(X) \tag{7}$$

The Randomised version of the Principal Component Analysis (RPCA) is a dimensionality reduction method, which uses approximated Singular Value Decomposition. RPCA offers efficiency improvements over alternative SVD solutions when applied to large scale problems (Bjarkason et al., 2018; Halko et al., 2009). The goal of this approach is to reduce the computational load on the low power devices running the ML algorithm. For developing a lightweight machine learning approach. By applying RPCA it reduces the computational time further.

The primary purpose behind using the PCA method in the approach depicted in ***Figure 5*** is to transform the *n* rows of the input feature space **X** of dimensionality *p* (**X** is a *n* by *p* matrix) into a space of reduced dimensionality **Y** where **Y** is a *n* by *l* matrix with $l < p$ by means of a linear transformation defined through the weights **w** **(8)**. Third equation of **(8)** shows the mapping of the i-th row of X, x(i) into the corresponding principal component **y:**

$$
\begin{aligned}
&\mathbf{w}_{(k)} = (w_1, \dots, w_p)_{(k)} \\
&\mathbf{y}_{(i)} = (y_1, \dots, y_l)_{(i)} \\
&y_{k(i)} = \mathbf{x}_{(i)} \cdot \mathbf{w}_{(k)} \qquad \text{for} \qquad i = 1, \dots, n \qquad k = 1, \dots, l
\end{aligned} \tag{8}
$$

The first component of the PCA is calculated such that variance is maximized by forming and maximizing the Rayleigh quotient **(8)**. The maximum of the Rayleigh quotient **(9)** is the largest eigenvalue of the matrix $\mathbf{X}^T\mathbf{X}$ occurring when **w** is the corresponding eigenvector.

$$
\mathbf{w}_{(1)} = \arg\max\{\frac{\mathbf{w^T X^T X w}}{\mathbf{w^T w}}\} \tag{9}
$$

The *k*-th component returned by the PCA transformation is calculated by subtracting the *k-1* components **(10)** from **X** and the corresponding weights of the residual matrix similarly to the first component **(11)**.

$$
\hat{\mathbf{X}}_k = \mathbf{X} - \sum_{s=1}^{k-1} \mathbf{X}\mathbf{w}_{(s)}\mathbf{w}_{(s)}^{\mathrm{T}} \hat{\mathbf{X}}_k = \mathbf{X} - \sum_{s=1}^{k-1} \mathbf{X}\mathbf{w}_{(s)}\mathbf{w}_{(s)}^{\mathrm{T}} \tag{10}
$$

$$
\mathbf{w}_{(k)} = \arg\max\{\frac{\mathbf{w^T \hat{X}_k^T \hat{X}_k w}}{\mathbf{w^T w}}\} \tag{11}
$$

From equation **(8)** it can be seen that PCA is not only reducing the dimensionality of the feature space but also mixing the information carried by the initial set of features, as opposed to feature elimination approaches, which would just drop one or more columns of the matrix **X**. For this reason, the method based on PCA is in the general case adding the flexibility, which allows more accurate dimensionality reductions.

To present the proposed method more formally, it can be defined with the operators that return the reduced feature space by eliminating features either using VIF or RFE **(12-14)** and the reduced feature space obtained by extracting principal components **(14)**, when applied to the non-reduced input space X:

$$X_{red,VIF} = \Psi_{red,VIF}(X) \tag{12}$$

$$X_{red,RFE} = \Psi_{red,RFE}(X) \tag{13}$$

$$X_{red,PCA} = \Theta_{PCA}(X) \tag{14}$$

Furthermore, it is convenient to define the following Boolean expressions that take the value 1 when the condition of the coefficients are verified and 0 when it is not verified. The Boolean expression **(15)** carries the value 1 when the VIF method must be applied according to the flow of ***Figure 5*** and the value 0 when the VIF method is not used. The complementary Boolean expression **(16)** takes the value 1 when the RFE method is applied and 0 otherwise.

$$\max(c_{ij}) > c_{max} \tag{15}$$

$$\max(c_{ij}) \leq c_{max} \tag{16}$$

With the definitions given above, the proposed automated feature selection method can be expressed as the following equation **(17),** to evaluate the reduced output feature space X_{red}:

$$\begin{aligned} X_{red} = \Theta_{RPCA}\{[\max(c_{ij}) > c_{max})]\Phi_{red,VIF}(X) \\ +[\max(c_{ij}) \leq c_{max})]\Phi_{red,RFE}(X)\} \end{aligned} \tag{17}$$

Chapter 4: Initial Experiment using the UCI Machine Learning Repository

4.1 Common ML Problems

Instead of directly theorising how various ML models should operate and adapt to numerous datasets, this case study demonstrates, through experimental evaluation, the effectiveness of the new method's lightweight capabilities. The initial case study also employs various ML models appropriate and effective at solving different problems of economic and scientific interests, evaluating the models through mathematical and experimental analysis using datasets from the UCI Machine Learning Repository (Dheeru & Karra Taniskidou, 2017). Ultimately, the focus of this experiment is understanding quickly what works and what doesn't work in improving the efficiency of the ML model so that the findings can pave the way for exploring more complex challenges.

The characteristics of machine learning are the use of algorithmic techniques to enhance descriptive, predictive and prescriptive efficiencies in real-world situations (Huddleston & Brown, 2019). There are three main types of machine learning algorithms; Supervised Learning, Unsupervised Learning and Reinforcement Learning. Reinforcement Learning exposes the algorithms to an environment in which it uses trial and error to train itself continuously. The model learns from its experiences and attempts to best understand data presented to it to make appropriate decisions. Unsupervised Learning algorithms attempt to identify hidden structures in data. This process of clustering populations into structures of different groups without a given target variable is referred to as being unsupervised. However, this study focuses on Supervised ML techniques as the decision boundaries can be clearly defined for the purpose of the lightweight ML application use case for PA monitoring. Supervised ML consist of a labelled target variable y that is to be predicted as a target function f from a given set of observations or predictor variables x. Supervised Learning algorithms are subdivided into *regression* and *classification* (which is further subdivided into *binary* and *multinomial classification*) problems (Mehra & Gupta, 2013).

This experiment aimed to establish the following:

1. To apply a lightweight ML model to both regression and classification problems to validate the performance.
2. To identify what type of problem is the new lightweight ML model best suited for.
3. Establish the feasibility of the approach and its application on low powered devices

4.2 The Experiment

In this work H2O, TPOT, and Auto-Sklearn Python libraries together with various public datasets, have been used for understanding the automated ML pipeline. These tools can help to quickly evaluate feature constructions, feature selections, model selection, and hyperparameter tuning configurations for data scientists to begin unravelling patterns from within their data. It allows data practitioners to focus on more critical aspects of the process.

The proposed ML approach was evaluated by handling three types of ML problems, binary and multi-class classification and regression problems. The performance of three automated machine learning libraries was recorded as the baseline model. The baseline model is the ML carried out on the Cloud, which has significantly more computational resources. The baseline model provides a mechanism for producing a benchmark (Schwab & Starbuck, 2013) that can be used for comparing the ML performance between the baseline and the ML running on the low powered device. To study and evaluate the effectiveness of the proposed method. Fourteen datasets from the UCI Machine Learning Repository (Dheeru & Karra Taniskidou, 2017) were selected, as presented in ***Table 2***, most of which were used to evaluate AutoML (Feurer et al., 2015a) and TPOT (Olson et al., 2016). The selected datasets comprising of four binary classification problems, six multi-class problems and four regression problems were used in the experiment to demonstrate the broader capabilities of the approach.

Table 2. A summary of the UCI datasets used for experimentation and their characteristics

Name	Task	Instances	Features	Missing Values
Adult Salary	*Binary-Class*	*48842*	*14*	*Yes*
German Credit	*Binary-Class*	*1000*	*20*	*No*
Breast Cancer	*Binary-Class*	*569*	*32*	*No*
Mushroom	*Binary-Class*	*8124*	*22*	*Yes*
Yeast	*Multi-Class*	*1484*	*8*	*No*
Amazon Reviews	*Multi-Class*	*1500*	*10000*	*No*
Car Evaluation	*Multi-Class*	*1728*	*6*	*No*
Abalone	*Multi-Class*	*4177*	*8*	*No*
Wine Quality	*Multi-Class*	*4898*	*12*	*No*
Waveform V2	*Multi-Class*	*5000*	*40*	*No*
Boston Housing	*Regression*	*506*	*13*	*No*
Cycle Power Plant	*Regression*	*9568*	*4*	*No*
Air Quality	*Regression*	*9358*	*15*	*Yes*
Energy Efficiency	*Regression*	*768*	*8*	*No*

4.3.3 Experimental Methodology

ML algorithms are stochastic in nature and require random resampling to minimise the effects of confounding variables and to achieve a generalised performance score. This means it is essential to perform multiple runs of the same experiment to control the source of variance. To control the variations in the results, each dataset was run ten times with a different random seed to randomly split the datasets as; 70% for training and 30% for testing. To establish a baseline performance score for each library, TPOT, H2O and Auto-Sklearn, they were each run ten times, and their mean score was recorded. The default settings for TPOT were set as; 20 generations and 25 population size (500 pipelines) and evaluated with a 5-fold cross-validation. H2O, TPOT and Auto-Sklearn were configured to run for a maximum run time of 30 minutes for each cycle and the memory limited to 256GB. Once the baseline results were all captured, an additional 10 minutes was allowed for the methods to complete feature engineering and feature selection before feeding the three libraries. For binary classification problems, the cross-entropy / log loss value was recorded. For multi-class problems, the zero-one loss values were recorded. For regression problems, the mean absolute error was recorded.

4.3.4 Results and Discussion

During the initial experimental evaluation, the binary classification datasets were examined first. Then proceeded to compare the results using the computed mean log loss value. ***Table 3*** shows the comparative percentage performance gains and loses across three auto ML libraries to their baseline, and the bold figures highlight where there were positive gains in performance. The binary cross-entropy / log loss measure was used to evaluate the performance for this binary classification experiment.

Table 3. Shows the comparison of the mean log loss, the time taken to complete in brackets and the % performance gain/loss made on the binary classification datasets

Datasets	**Baseline Model log loss & (time)**	**New Approach log loss & (time)**	**Log loss % difference**	**% Time reduced**
Adult Salary	*0.191 (42.80s)*	*0.199 (12.42s)*	***-4.19%***	***71%***
German Credit	*0.568 (6.32s)*	*0.594 (2.84s)*	***-4.58%***	***55%***
Breast Cancer	*0.187 (4.05s)*	*0.191 (2.23s)*	***-2.14%***	***45%***
Mushroom	*0.568 (18.07s)*	*0.674 (6.87s)*	*-18.66%*	*62%*

Table 3 shows that two out of four datasets showed that the model was able to achieve a log loss within 5% of the baseline. The columns titled "*Baseline Model log loss (time)*" and "*New Approach log loss (time)"* present the mean log loss and the average time in seconds it took to complete the training and classification denoted within brackets. The "*Log loss % difference*" column provides the percentage difference between the log loss values recorded for each model. The "*% Time reduced*" column provides the percentage difference in the time taken to complete the two models' training and classification. On initial observations, the four datasets, have larger data sample increasing the chances of identifying suitable features to eliminate and have more than twenty original features to choose from. With more features, the probability of finding monotonic relations between them grows, which is evident in the correlation matrix in ***Figure 6***.

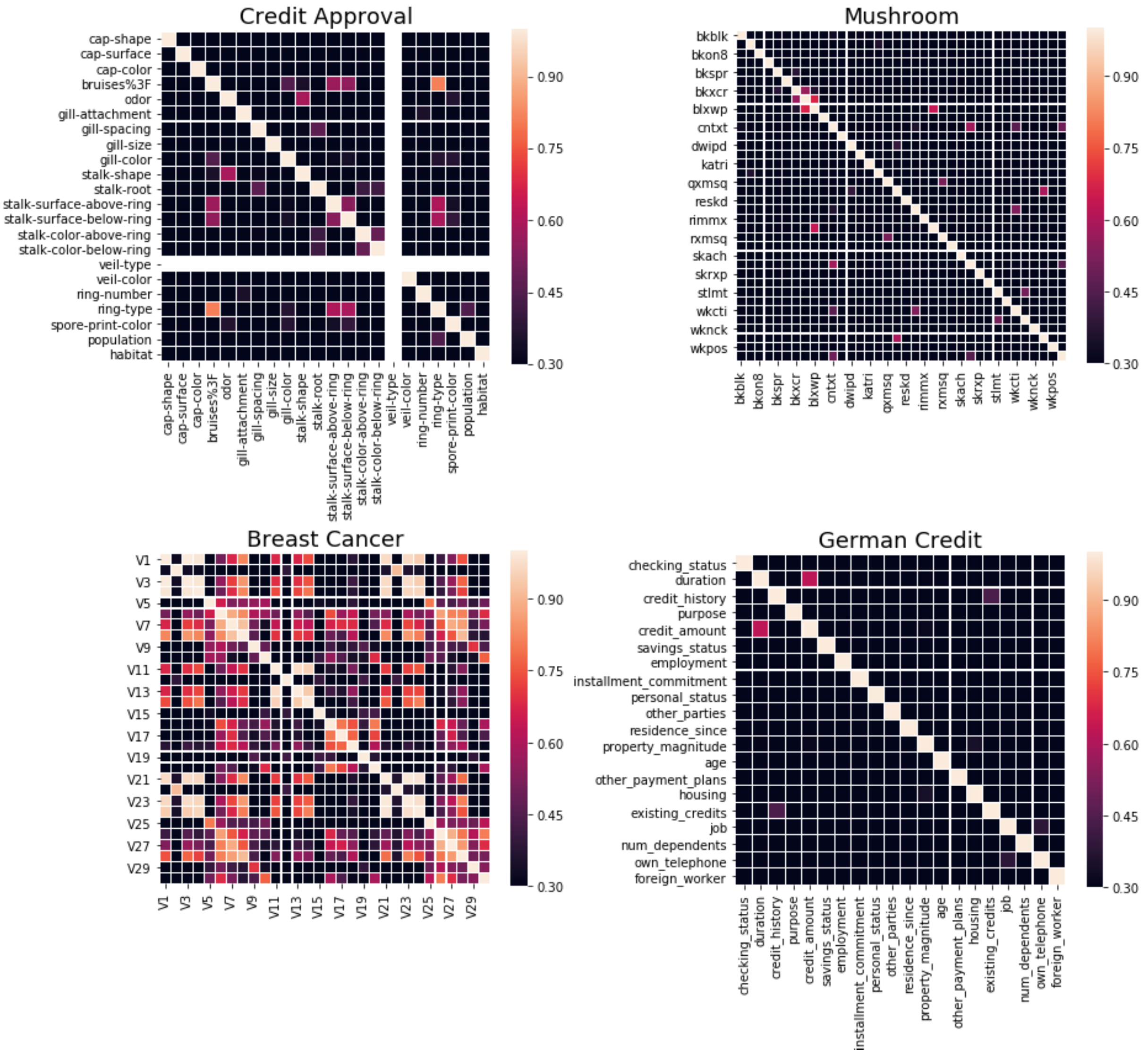


Figure 6. Shows the correlation matrix of the four datasets

Figure 6 shows the correlation matrix of four datasets and three (Credit Approval, Mushroom, and Breast Cancer) of which were suitable for the VIF feature evaluation. The other datasets did not produce a suitable correlation matrix instead of the remaining three (Adult Salary and German Credit) datasets we suitable for RFE method. The correlation coefficient values between the features were below the threshold, which provided a performance score within 5% of the baseline model.

Figure 7 shows the feature selection plots after evaluating the elements within the datasets. The plots show that it is possible to eliminate some features without losing significant performance. However, the Mushroom dataset did perform the worst out of the four datasets and performed well below 15% of the baseline.

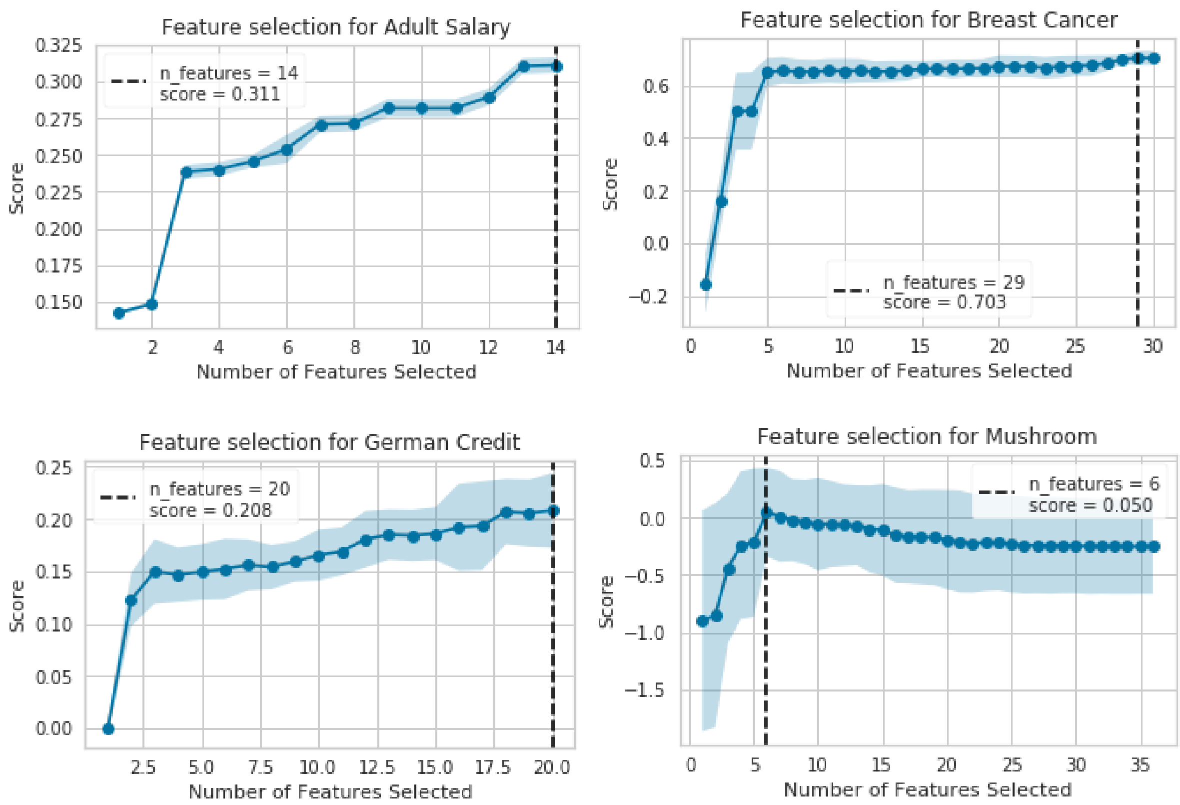


Figure 7. Feature selection performance for binary classification datasets

The method successfully identified suitable features, during the evaluation stage, and was able to filter out specific elements that could reduce the model size while improving the efficiency of the ML model. ***Figure 8*** depicts the percentage log loss distribution, from the baseline, of the datasets that positively responded to the ML efficiency improvements, as shown in **Figure 9**, while maintaining the performance within 5% of the baseline. It also shows the efficiency gains of above 45% across all four datasets running on the lower power device.

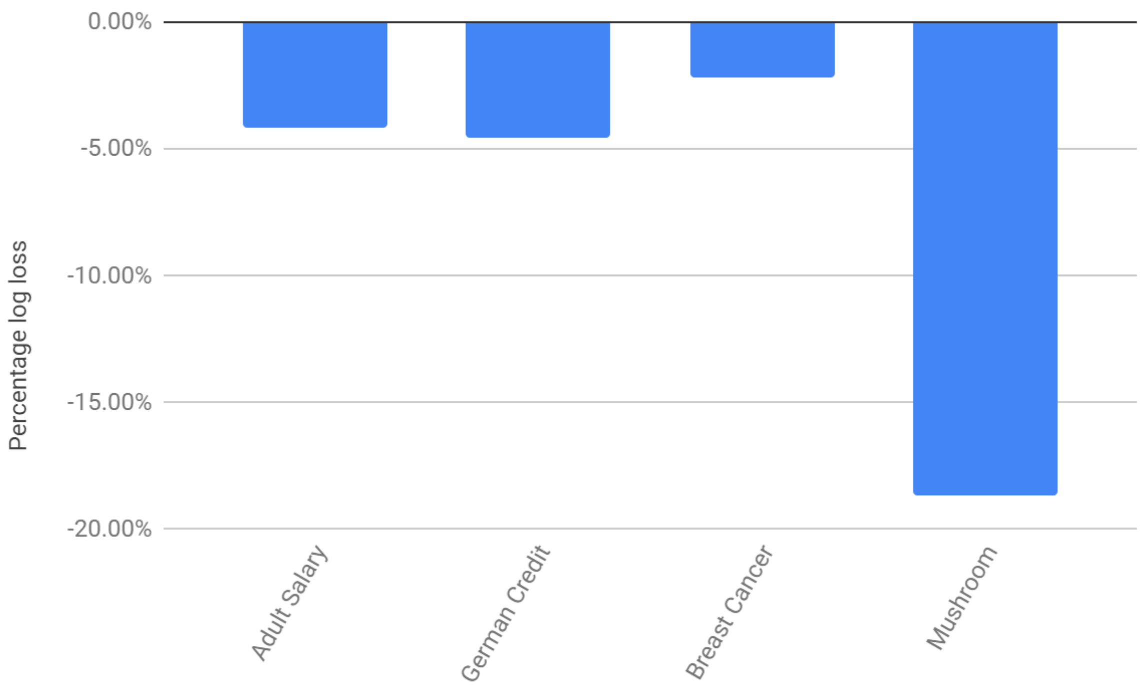


Figure 8. The percentage log loss for binary classification datasets from the baseline model

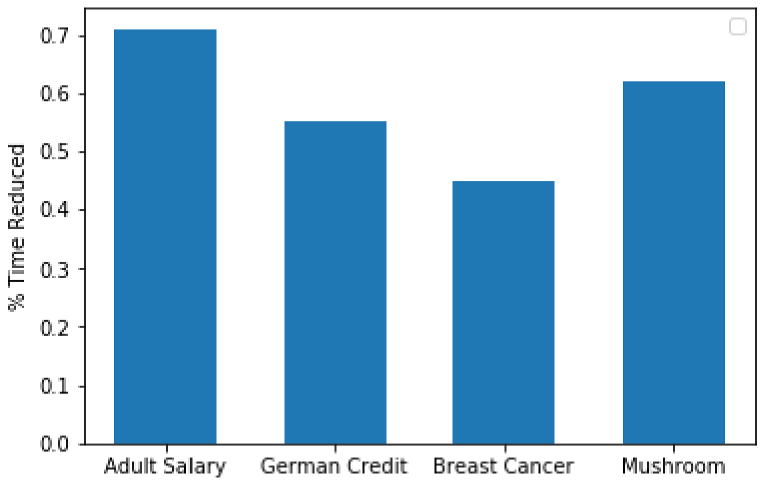


Figure 9. The percentage efficiency gains for binary classification from the baseline model

Table 4 presents the results of the experimental evaluation on the multi-class dataset. The table shows the mean zero-one loss value for each dataset and comparative performance variance in gains and or losses against the respective baseline. The figures in bold again highlight where there were performance gains. It shows that datasets that can maintain their performance within 5% of the baseline after reducing the data model's size: *Amazon Reviews* and *Waveform V2*. The remaining datasets were unable to determine any monotonic correlations amongst the predictor variables or find any suitable features to eliminate. The results indicate the proposed method can effectively help improve the performance of the multi-class classification problems. However, the data sample size and the number of original features play a role in identifying suitable elements for feature reduction. ***Figure 10*** shows the feature selection across the datasets with the increase in performance. It also shows that the Amazon Reviews and Waveform V2 are best suited in benefiting from the proposed approach. The remaining datasets do not show significant gains in performance by filtering featuring.

Table 4. Shows the zero-one loss, time taken in seconds in brackets and % gain/loss (gains are in bold) for multi-classification

Datasets	Baseline Model zero-one loss & (time)	New Approach zero-one loss & (time)	Zero-one Loss % difference	% Time reduced
Yeast	*0.416 (11.03s)*	*0.498 (4.30s)*	*-19.71%*	*61%*
Amazon Reviews	*0.567 (16.70s)*	*0.569 (5.84s)*	*-0.35%*	*65%*
Car Evaluation	*0.015 (8.04s)*	*0.019 (3.93s)*	*-26.67%*	*51%*
Abalone	*0.758 (12.16s)*	*0.848 (4.98s)*	*-11.87%*	*59%*
Wine Quality	*0.245 (18.76s)*	*0.269 (5.25s)*	*-9.80%*	*72%*
Waveform V2	*0.102 (20.38s)*	*0.109 (7.95s)*	*-6.86%*	*61%*

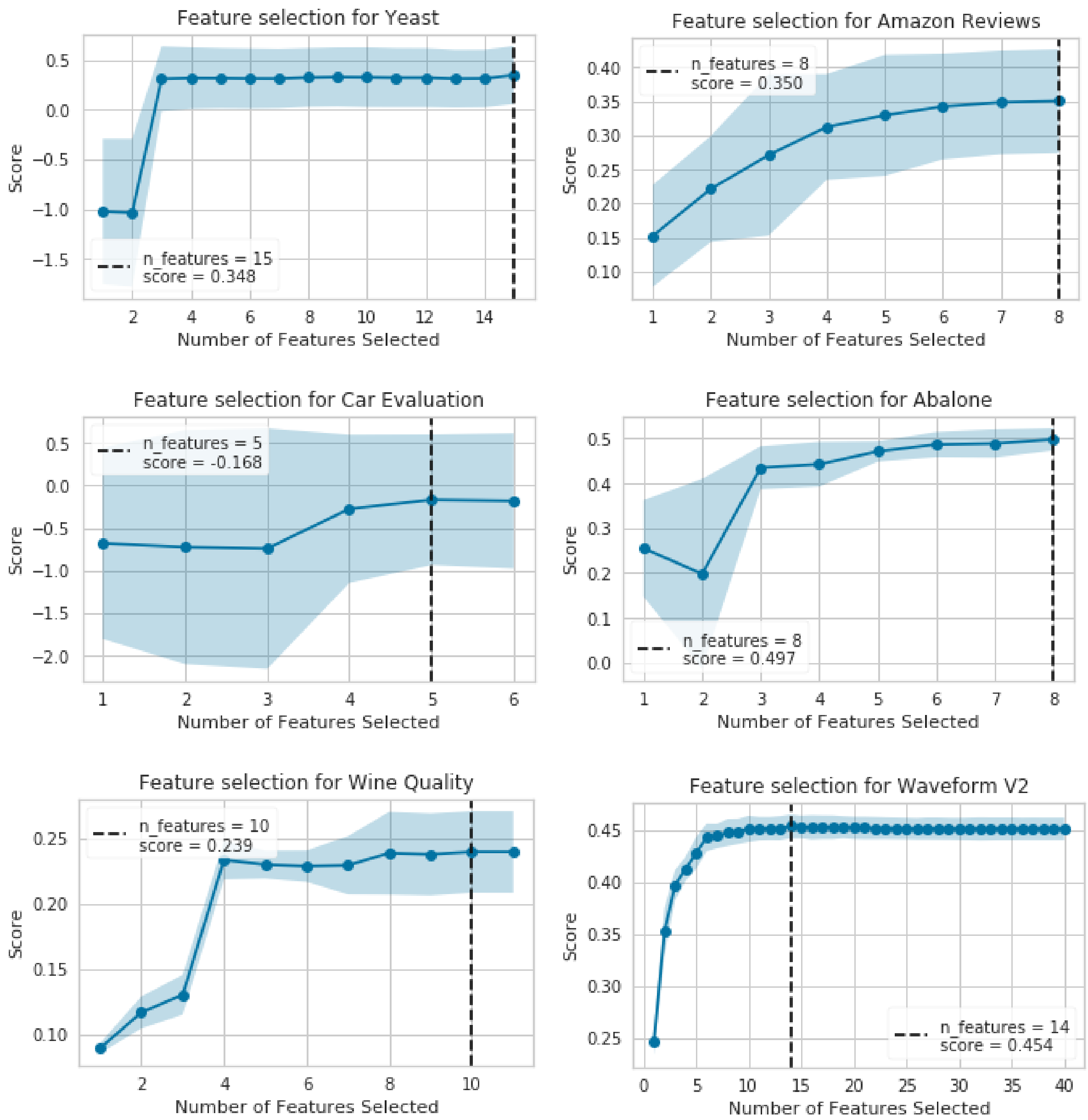


Figure 10. Feature selection performance of the multi-class classification datasets

Figure 11 below shows that the distribution of performance, results mapped from *Table 3,* that most datasets in the multi-class group have benefitted from the proposed method. The datasets with feature pairs correlated have been removed, reducing the model size and improving their efficiency. The datasets such as Yeast, Care Evaluation, Abalone and Wine Quality contained a small sample size, and all had a minimal number of features, which reduced the probability of identifying any pairwise feature correlations and hence performed significantly lower than 5% of the baseline. It would be recommended that a standard ML approach is used for these datasets.

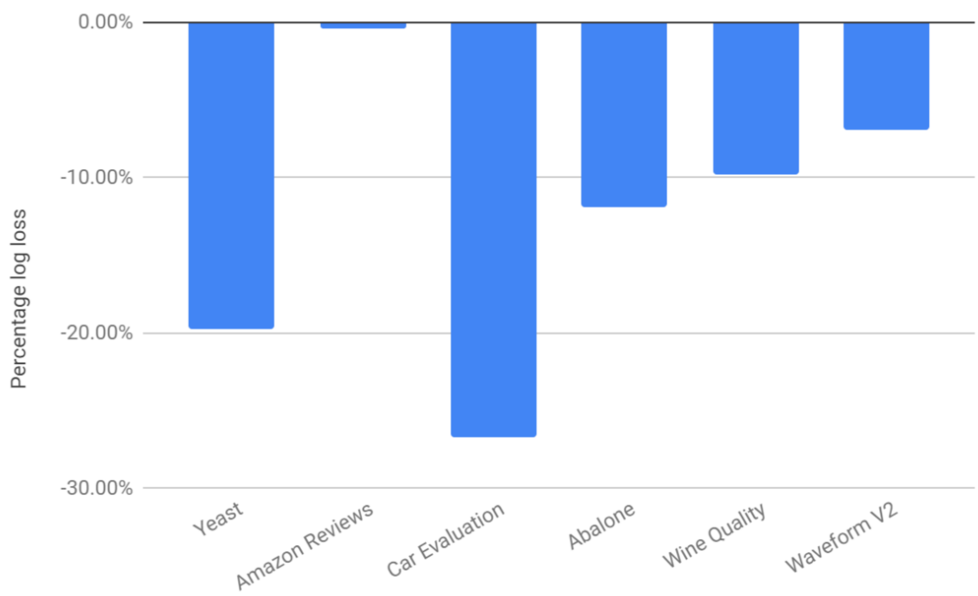


Figure 11. Shows the percentage gain distribution across the multi-class datasets

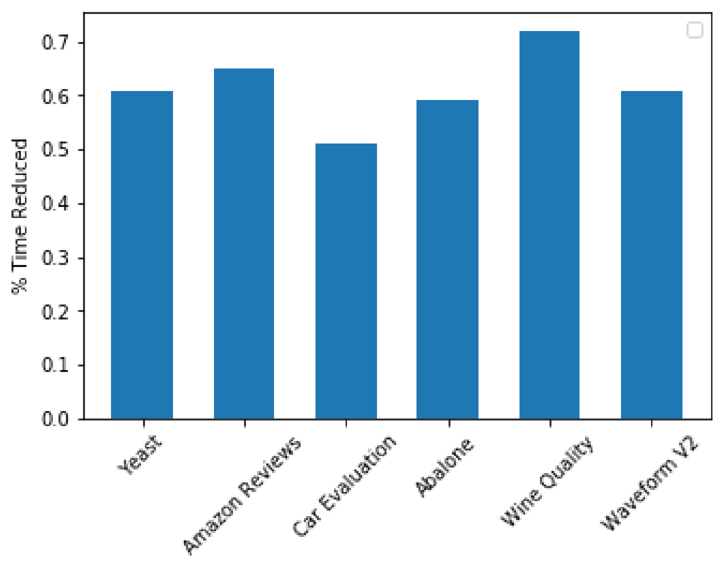


Figure 12. The percentage efficiency gains for binary classification from the baseline

Figure 12 shows that efficiency gains above 50% was achieved across all the datasets as the model sizes were reduced to run on the lower-powered device.

Finally, ***Table 5*** shows the results of the experimental evaluation carried out on regression problems. The table shows the mean of the ten mean absolute error (MAE) values recorded after each run. The *Boston Housing and Air Quality* both had more than ten features with reasonable sample sizes and performed well during testing.

Table 5. Shows the mean absolute error (MAE), the time taken in seconds in brackets and the % performance gain/loss for regression

Datasets	**Baseline Model MAE & (time)**	**New Approach MAE & (time)**	**MAE % difference**	**% Time reduced**
Cycle Power Plant	*2.474 (21.69s)*	*2.998 (6.29s)*	*-21.18%*	*71%*
Boston Housing	*2.533 (8.73s)*	*2.665 (5.15s)*	*-5.21%*	*41%*
Energy Efficiency	*1.186 (7.33s)*	*1.483 (4.10s)*	*-25.03%*	*44%*
Air Quality	*2.470 (23.78s)*	*2.588 (6.18s)*	*-4.78%*	*74%*

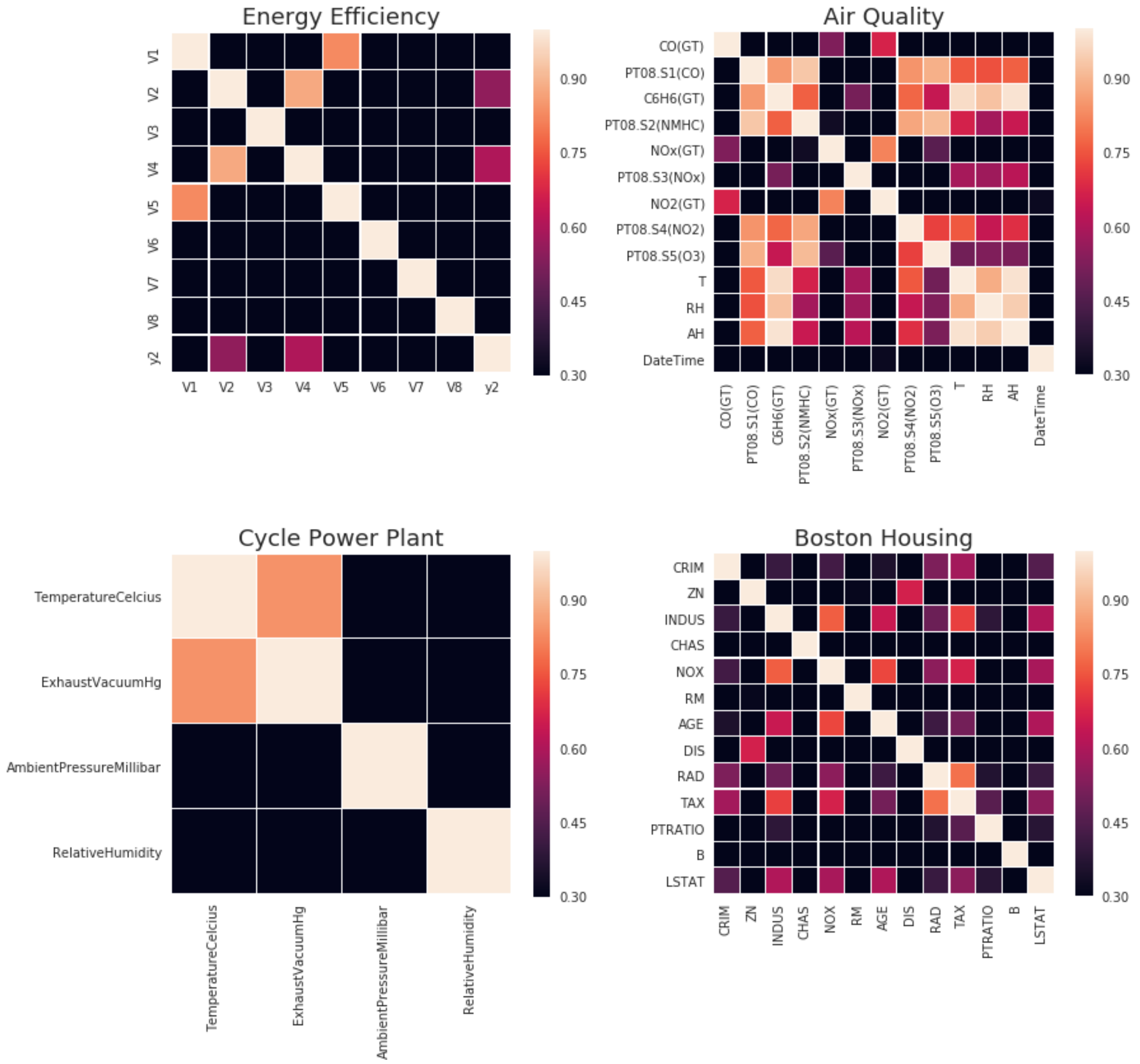


Figure 13. Shows the correlation matrix of the four regression datasets

Figure 13, shows the correlation matrix of the four regression datasets evaluated to determine the effectiveness of the approach. The Air Quality and Boston Housing datasets both had feature correlations above the 0.5 initial thresholds. Features were then filtered out using the VIF method and further reduced using RPCA to achieve a mean absolute error within 5% of the baseline. ***Figure 14,*** shows the distribution of performance gains across the regression datasets. It suggests that the regression problems can benefit from improving how features are engineered and selected using the proposed method. However, where there are fewer features, and the sample size was small, the performance did suffer when using this new approach. The results indicate that the approach favours linear data structures where the data is sequential and predictable. ***Figure 15,*** shows that efficiency gains above 40% were achieved across all the datasets as the model sizes were reduced to run on the low powered device.

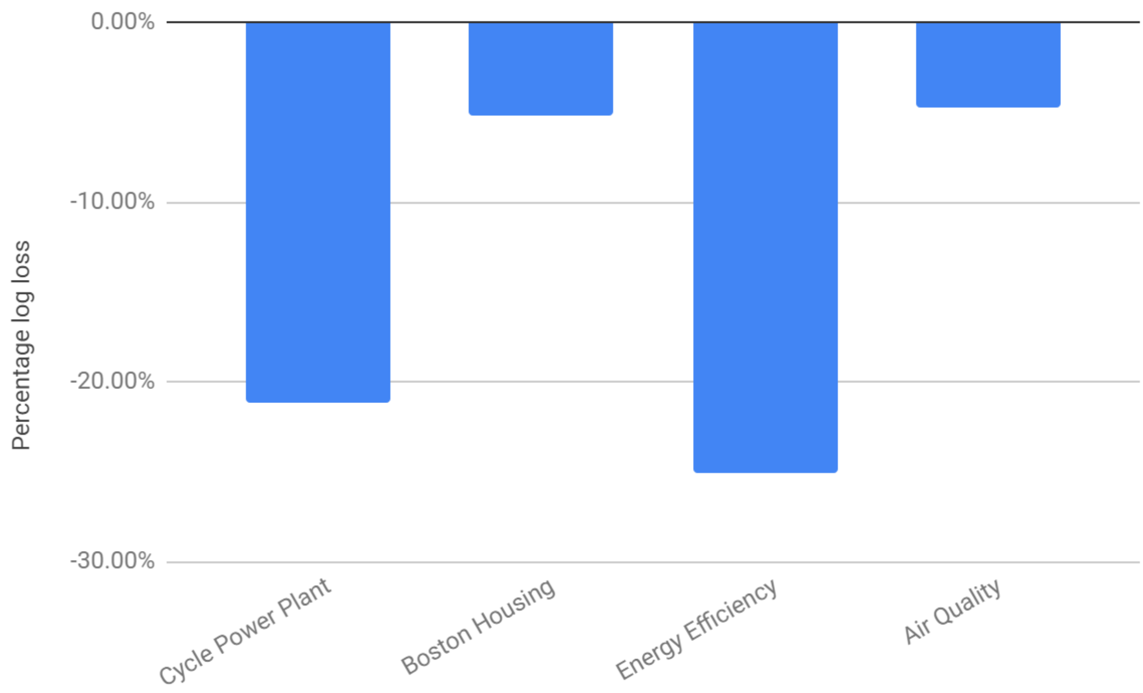


Figure 14. Shows the percentage gain distribution across the regression datasets

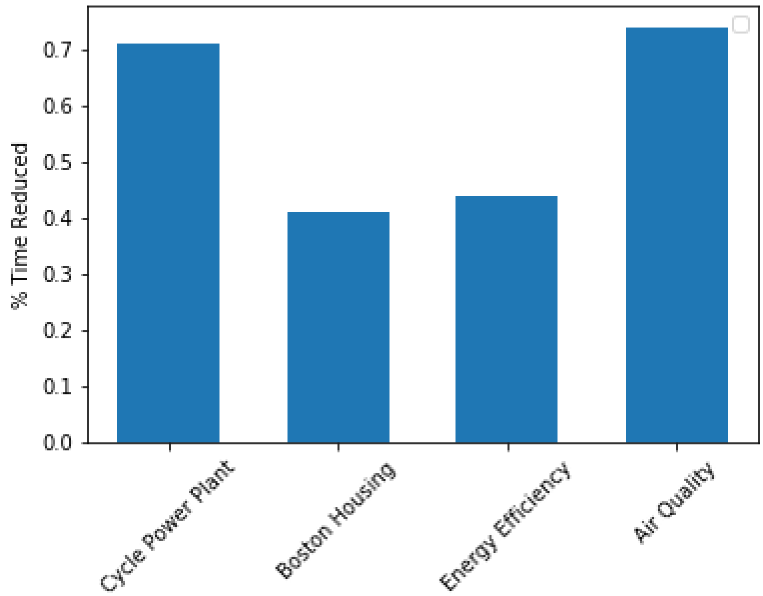


Figure 15. The percentage efficiency gains for regression models from the baseline model

4.4 Conclusion

This study empirically demonstrated that sufficient gains could be achieved and the proposed approach performed exceptionally well with high dimensional datasets where features can be eliminated due to their monotonic relationships between them. As a result, reducing the model size and increasing the model's efficiency. The experimental results on the UCI public data sets confirm that the proposed approach is only practical on data sets which contain a large enough sample size with features with monotonic correlations that can be eliminated. The results show that it can achieve similar reasonable performance within 5% of the baseline model in certain use cases while running on lower-powered devices. The results also suggest that the proposed approach benefits situations that fit their specific criteria where there are correlated features. It has also proved to apply to both regression and classification problems. For binary classification, 45% efficiency gains were achievable, for multi-class classification, 50% efficiency gains were possible, and finally, 40% efficiency gains were likely for regression problems. However, the actual performance results indicate that effective practice will require significant improvements to enable this approach to be suitable for use cases, linear in nature.

Chapter 5: Case Study One - Predicting Children's Energy Expenditure during Physical Activity using Deep Learning on a Low Powered Device

Low Powered Devices (LPD) such as wearable accelerometers and raspberry pi3 are the widely used tool to assess physical activity (PA) to assess energy expenditure and calculate the time spent during intense PA (Crouter et al., 2018). Studies in PA assessment are increasingly using accelerometry data to estimate PA in children (Crouter et al., 2018; Alex V. Rowlands et al., 2013). Such methods provide objective data, given the difficulties in assessing PA in children due to the inherent more significant movement variability in children than adults. Considerable effort has been made to standardise accelerometer derived PA data which is becoming essential to accurately estimate PA in paediatric populations (Roscoe et al., 2017; Duncan et al., 2016; Phillips et al., 2013; Ryan & Gormley, 2013). In this case study, the proposed lightweight ML approach, including the auto feature selection is deployed on the raspberry pi3 to model the PA in children to monitor energy expenditure.

This study has the following two contributions:

- Carry out the ML process on the LPD (e.g. raspberry pi3) including the auto feature selection as part of the ML process.
- Accurately predict the energy expenditure value up to 90% by using the proposed lightweight ML model

5.1 Experimental Setup

5.1.1 Participants

The data was supplied in collaboration with the Faculty Research Centre for Sport, Exercise and Life Sciences at Coventry University. The participants included 28 healthy Caucasian children aged between 8 and 11 years of age. This sample included 13 girls and 15 boys (Mean ± SD = 9.4 ±1.4 years). All the participants were from central England who took part in this study. All of whom followed the institutional ethics approval process. The children's parental written and informed consent and the child's permission was obtained. The participants Mean ± SD of height, mass and body mass index (BMI), was 1.4 ± 0.4m, 34.6 ± 8.6 kg and 17.6 ± 2.5 kg/m2, respectively.

5.1.2 Procedures

Participants wore a GENEActiv monitor (Activinsights, Cambridgeshire, UK) on both wrists and waist, similar to work carried out by Routen et al. (2012), an additional monitor was also fitted around the dominant ankle. The devices were worn continuously throughout the testing period. The GENEActiv device was configured to record at 80Hz at 1-second epochs.

Each participant was required to performed a series of activities reflecting the different levels of PA. All activities lasted for five minutes with five-minute rests in between.

5.1.3 Data Processing

Once all the testing was completed, each participant's wearable accelerometer and calorimetry data was transferred to the low powered device (raspberry pi3). The first and last minute of each 5-minute activity was discarded, leaving 3 minutes for analysis. To ensure that MET values for each activity were at the required intensity and consistent with work carried out by Roscoe et al. (2017) and Phillips et al. (2014). The raw data for each wear location were summed into a signal magnitude vector (gravity subtracted) expressed in 1s epochs, based on similar methods employed by Esliger et al. (2011) and Phillips et al. (2013).

The actual target values, the Metabolic Equivalents (MET), were calculated from the VO_2 values, the maximum amount of oxygen utilised during exercise measured in millilitres of oxygen consumed in one minute (Duncan et al., 2019), per kilogram of body weight (mL/kg/min), using age-specific values (Harrell et al., 2005) and categories into one of four intensities Sedentary (< 1.5 METs), Light (1.5-2.99 METs), Moderate (3-5.99 METs) and Vigorous (>6 METs).

The input features captured for data processing were broken down and converted into numerical notation. The categorical input feature the type of activity performed categorised as Sedentary (Sed), Light, Moderate (Mod or Vigorous (Vig), was one-hot encoded. The gender was integer encoded as 0 for male and 1 for female. The Moderate to vigorous physical activity (MVPA) was integer encoded as 0 for false and 1 for true. The sensor values captured by the wearable were as follows: the non-dominant wrist as GeneaCount_NonDom_Wrist, the value captured on the dominant wrist as GeneaCount_Dom_Wrist, the value captured from the waist as GeneaCount_Waist and the dominant ankle as GeneaCountDomAnkle. Finally, the additional input parameters included the child's age during the test as Age_on_Test, the height in centimetres, the weight in kilograms, the child's sitting height in centimetres and the child's leg length in centimetres.

5.1.4 Statistical Analyses

Modelling algorithms all make assumptions, and the challenge here is to select an algorithm whose premises fit the current dataset and the modelling goals. If the dataset and or its feature transformations do not significantly deviate from the assumptions such as the Gaussian distribution test of the target variable, which helped indicate that a parametric approach was appropriate. For this reason, a Skewness and Kurtosis test was performed to help better understand the data by performing a normality test and establish whether a parametric approach is suitable for the current dataset. The analysis also helped identify the significance of the wear location and its impact on the predictive model used. To get a better perspective of the patterns from within the dataset four models (linear, ridge, lasso and a non-optimised neural network) were used in a heuristic approach to analyse the predicted values against a given sample to understand their potential capabilities from a baseline model. The approach provides a reference point from which to compare various machine learning algorithms and a means to measure performance changes. The approach has been incredibly effective, as demonstrated by Gjoreski et al., (2013), at producing a suitable baseline for comparison using similar regression models to predict the MET outputs.

Once the baseline was established, the proposed new approach can be run to evaluate the performance.

The steps followed were carried out:

1. Evaluate the threshold that achieves an accuracy score within 5% of the baseline.

2. Compute the correlation matrix and check whether there are any elements above or below the threshold.

3. As there were some highly correlated features in the correlation matrix, the VIF calculation was used to detect collinearity. Observations showed that some elements of the VIF array were substantial (in the order of *1e10*) compared to others that are in the order of *1e0* or *1e4*.

4. Define a threshold for the VIF feature reduction: *1e9*. The features (columns of the input dataset) whose index corresponds to the index of the VIF's array elements larger than the chosen threshold is eliminated.

5. RPCA is performed to further reduce the feature space dimension after eliminating collinear features, retaining 80% of the variance of the original dataset. Thus, also reducing the model size.

Based on the literature review the conventional models considered for the comparative study are: Linear Regression (LR), Ridge Regression (RR), Lasso (LS), Multi-Layer Perceptron (MLP), Random Forests (RF), Convolutional Neural Network (CNN) and the gradient boosted decision trees, XGBoost (XGB). The splitting between training and testing datasets has in the first instance been obtained extracting the testing datasets randomly. Furthermore, the PCA has been applied to further reduce the feature space.

5.2 Experimental Analysis

5.2.1 Feature extraction and selection

The first-order statistical features were initially calculated to reduce the dataset before training. The following first-order features were generated; the mean, median, maximum, minimum, standard deviation, the variance and the root mean square power. The statistical normally test skewness and kurtosis were performed to determine if the distribution deviates from the norm. To ensure the appropriate machine learning models are being applied, it is necessary to test and verify the dataset is either Gaussian-like or non-Gaussian like and instead use non-parametric statistical methods.

Skewness is defined as:

$$skewnes(X) = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] \tag{18}$$

Where μ is the mean, and σ is the standard deviation. The skewness measures the asymmetry of the distribution, which usually resembles a bell shape distribution. If the skewness is high, then it means the distribution is less symmetrical and can be either pushed to the left or the right.

Kurtosis is defined as:

$$kurtosis(X) = E\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] \tag{19}$$

The Kurtosis helps to measure if the distribution is either heavy or light-tailed to test for normality. A 5% tolerance was applied to determine if the distribution deviates from the norm and also how we should proceed with our further analysis work.

The input features used for evaluation are shown in **Table 6**. The target variable y (MET) is a numerical value that ranges from 1.0 to 7.0.

Table 6. The input features and the corresponding numbers

Features	Name
1	*Sed*
2	*Light*
3	*Mod*
4	*Vig*
5	*MVPA*
6	*Gender*
7	*GeneaCount_NonDom_Wrist*
8	*GeneaCount_Waist*
9	*GeneaCountDomAnkle*
10	*GeneaCount_Dom_Wrist*
11	*Age_on_Test*
12	*Height(cm)*
13	*Mass(kg)*
14	*SittingHeight(cm)*
15	*LegLength(cm)*

To improve the model efficiency, while maintaining its accuracy, the features were further analysed by evaluating their importance. The input features and their order of importance per activity were analysed separately to select the most suitable features. First, the Pearson correlation evaluation on the input features are performed, which indicates the level by which two features are linearly related. When the correlation coefficient is closest to 0, it would indicate that there was little or no correlation. A negative value showed a negative correlation and positive value would indicate a positive correlation. Using the greedy optimisations technique called the Recursive Feature Elimination (RFE) was used to find the most suitable subset of features for the given activity. By reviewing the attributes of the RFE output report reveals the feature rankings and their order of importance.

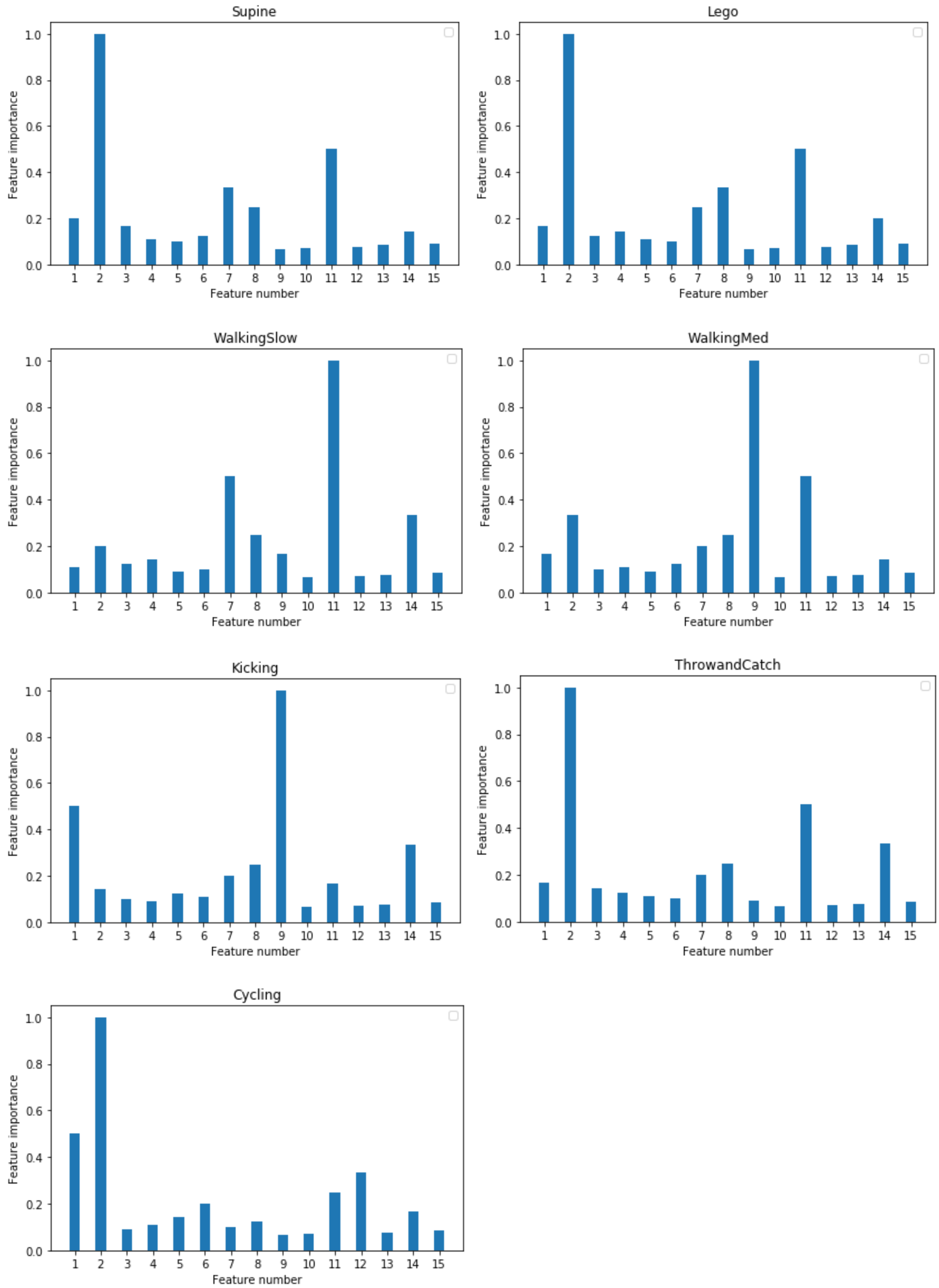


Figure 16. The inverse feature importance for each activity

Table 7. The input features and the corresponding numbers

Activity	***Features Selected***
Supine	[9,10,12,13,15,5,6,4,14,3,1]
Lego	[9,10,12,13,15,5,6,3,4,1,14]
Walking Slow	[9,10,12,13,15,5,6,1,3,4,9,2]
Walking Med	[10,12,13,15,5,3,4,14,6,1,7]
Kicking	[10,12,13,15,4,3,6,2,11,7,8]
Throw and Catch	[10,12,13,15,9,6,5,4,3,1,7]
Cycling	[9,10,13,15,3,7,4,8,5,14,6]

In ***Figure 16,*** it shows the inverse feature importance for the different activities included in the METs dataset, namely: *Supine*, *Lego*, *Walkingslow*, *Walkingmed*, *Running*, *Cycling*, *Kicking*. The Feature importance is depicted as the inverse of the ranking position in order of importance (the shorter the bar, the higher the importance). ***Table 6*** shows the feature labels that correspond to the numbers used to identify the features plotted in ***Figure 16***. ***Table 7*** shows the final feature features selected for each type of activity. The results suggest that if the key features are included with the accelerometer data, such as the type of activity, the gender, the mass, leg length and height as a minimum it was possible to achieve a high MET predictive score. However, removing this feature had little or no impact on the predictive performance of the model, as it always had other features as equally important such as the type of activity. Not knowing the type of activity (Sed, Light, Mod or Vig) did have an impact on the performance more so than any other feature.

5.2.1 Target Variable Analyses

The analysis of the target variable (METs) was undertaken to understand its meaning in order to proceed with modelling the dataset to predict the METs. ***Figure 17*** shows the MET frequency distribution and the probability quantile-quantile (Q-Q Plot). Skewness was also calculated to be 0.483084, suggesting the data was symmetrically distributed. Kurtosis was -0.168400 indicating there were little or no outliers.

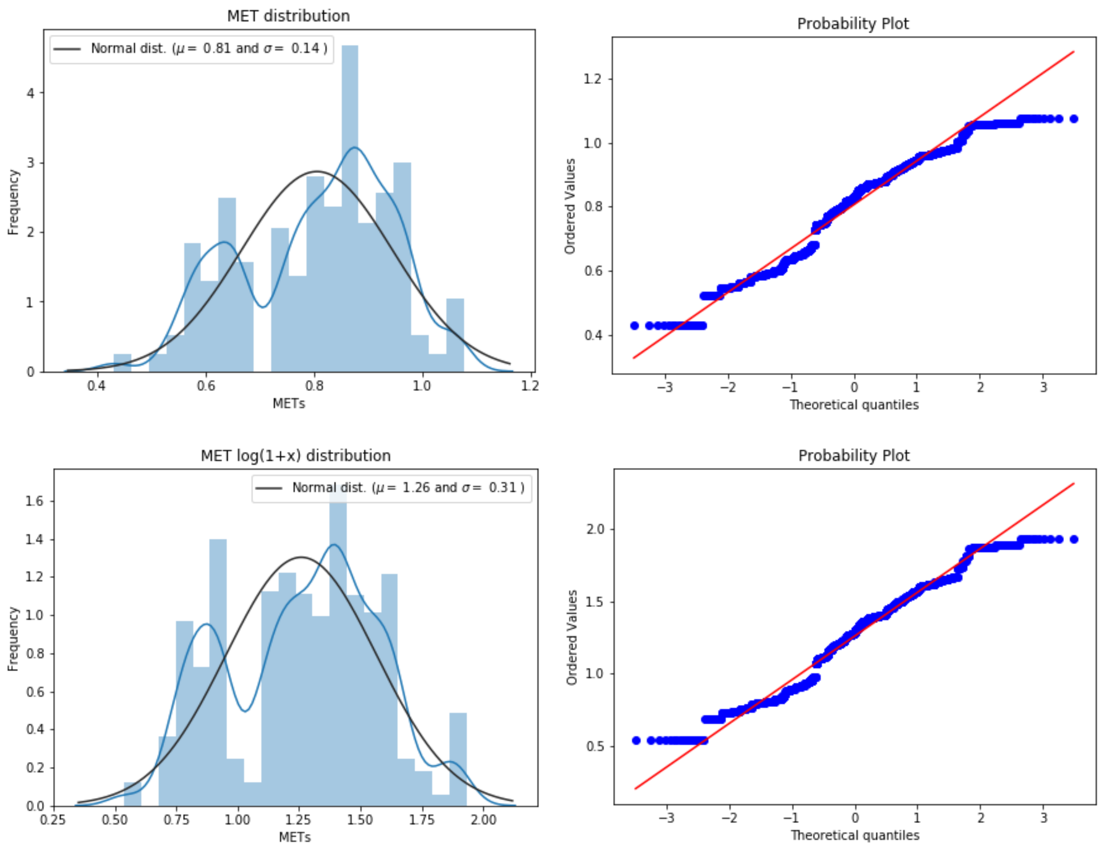


Figure 17. Frequency distribution and the probability quantile-quantile (Q-Q) plots for raw and log-transformed data

The MET values were skewed to the right, positive skewness $\mu = 2.69\ and\ \sigma = 1.11$, where the location parameters μ is the mean peak and σ is the standard deviation. Consequently, MET values were log-transformed by applying $log(1 + x)$ to all values. This transformation reduced the skewness $\mu = 1.26\ and\ \sigma = 0.31$. The theoretical quantiles and the linearity shown in ***Figure 17*** supports the fact the data is normally distributed. The Skewness value was also closer to 0 at -0.128513, and the Kurtosis value was -0.763134 supporting the earlier indicator of very few outliers.

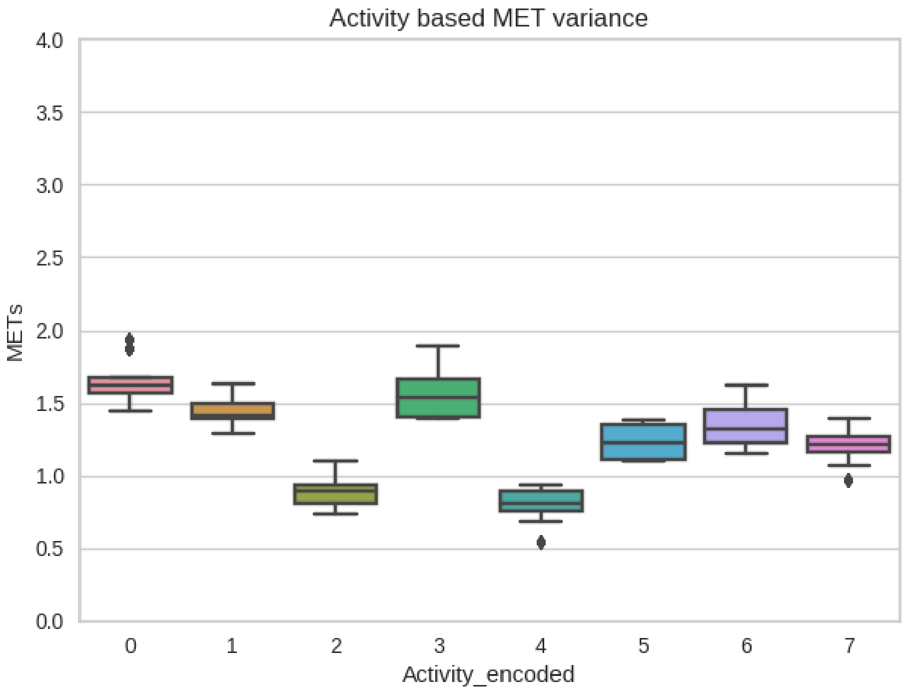


Figure 18. The relative MET variance of each activity

The box plot in ***Figure 18*** shows the 8 physical activities and their relative MET variance scores. The plot indicates very few outliners. The activity number and descriptions are shown in ***Table 8***.

Table 8. The encoded number associated with the physical activity

Number	Activity Description
0	*Cycling*
1	*Kicking*
2	*Lego*
3	*Running*
4	*Supine*
5	*ThrowandCatch*
6	*WalkingMed*
7	*WalkingSlow*

5.2.2 Predictor Variables Analyses

Given that the target variable had a Gaussian distribution to further improve the experimental models' performance a wrapper greedy optimisation algorithm was used to perform recursive feature elimination (RFE) to evaluate combinations of features and rank them based on the variables usefulness in improving the model's accuracy and the order of elimination. The weakest features were eliminated first, removing dependencies and collinearity that may exist, until an optimum subset was achieved that performed the best in cross-validation. The subset of features that scored the best was then used in further modelling.

Using a linear regression model, it was possible to identify the optimal feature subset as shown in ***Figure 19***. The plots show the number of features in a subset and the increasing accuracy of their performance with the optimal set being 20 features out of the given 24. The features at the top and bottom are the least colinear and are more important. Those at the centre indicate collinearity and are the least important. The Ridge regression model indicates there are at least 15 important features which can also achieve a potential accuracy of 85%, and the following subset of predictor variables were selected.

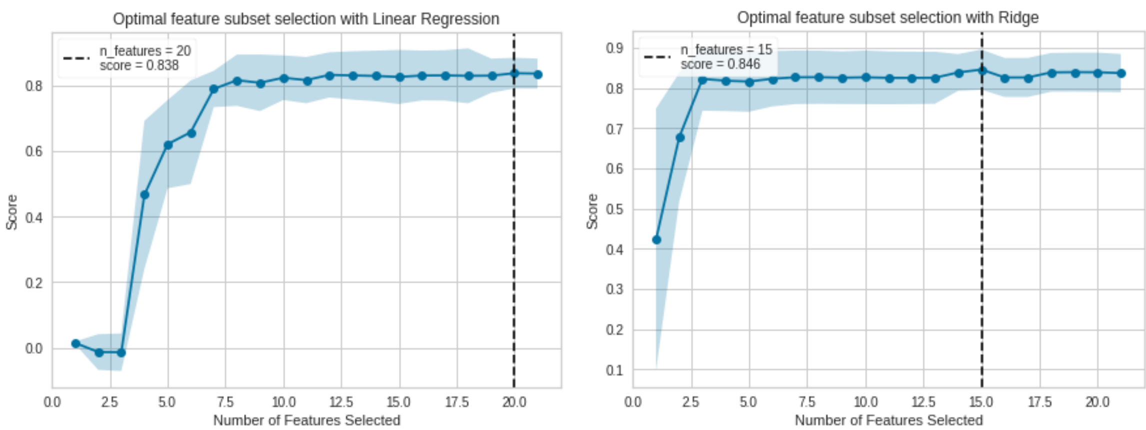


Figure 19. Optimal features subset selection

5.2.3 Results & Evaluation

Using the heat map plot shown in *Figure 20* supports the fact that the centre predictor variables are colinear and the least important and excluding them from the next stage of data modelling will improve the predictor performance.

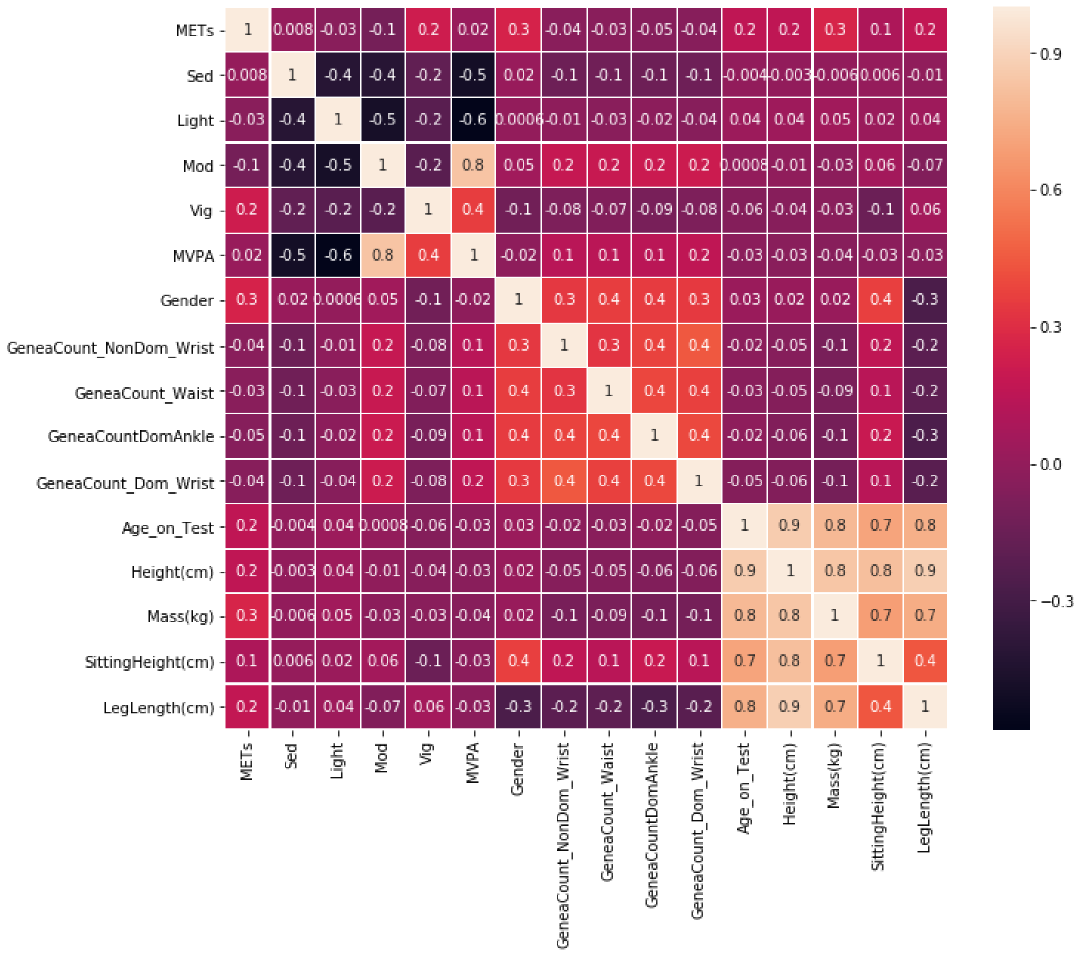


Figure 20. Correlation analysis for the MET activity dataset

There are many predictive modelling techniques to choose from, and selecting the right one is challenging. The simple approach is to evaluate their performance and or measure the impact of the wrong predictions. The performance of a model is often assessed by calculating the correlation coefficient or the regression of the model's predictions against the true values (Sheiner & Beal, 1981). To determine how much of the total variations in Y, the target variable MET, is expressed by the variations in X, the predictor subset variables and is defined as:

$$R^2 = 1 - \frac{\Sigma\left(Y_{actual} - Y_{predicted}\right)^2}{\Sigma(Y_{actual} - Y_{mean})^2} \tag{20}$$

The R^2 (Coefficient of determination) is used to measure how well the regression model approximates the actual values. In other words, how close the predicted data points fit the regression line (Barten, 1987). Values of R^2 range from 0 and 1, where 0 indicates no variability in the target variable and 1 indicates a perfect fit with its target. To set a baseline model, for simple regression based, predictive models were used to model the dataset using the selected features. Which includes, Linear Regression, Ridge Regression, Lasso Regression and a simple Multi-Layer Perceptron (MLP). The results of the computed R^2 measures and the prediction errors are shown in the plots in ***Figure 21,*** for each of the four models used in the initial baseline experiments.

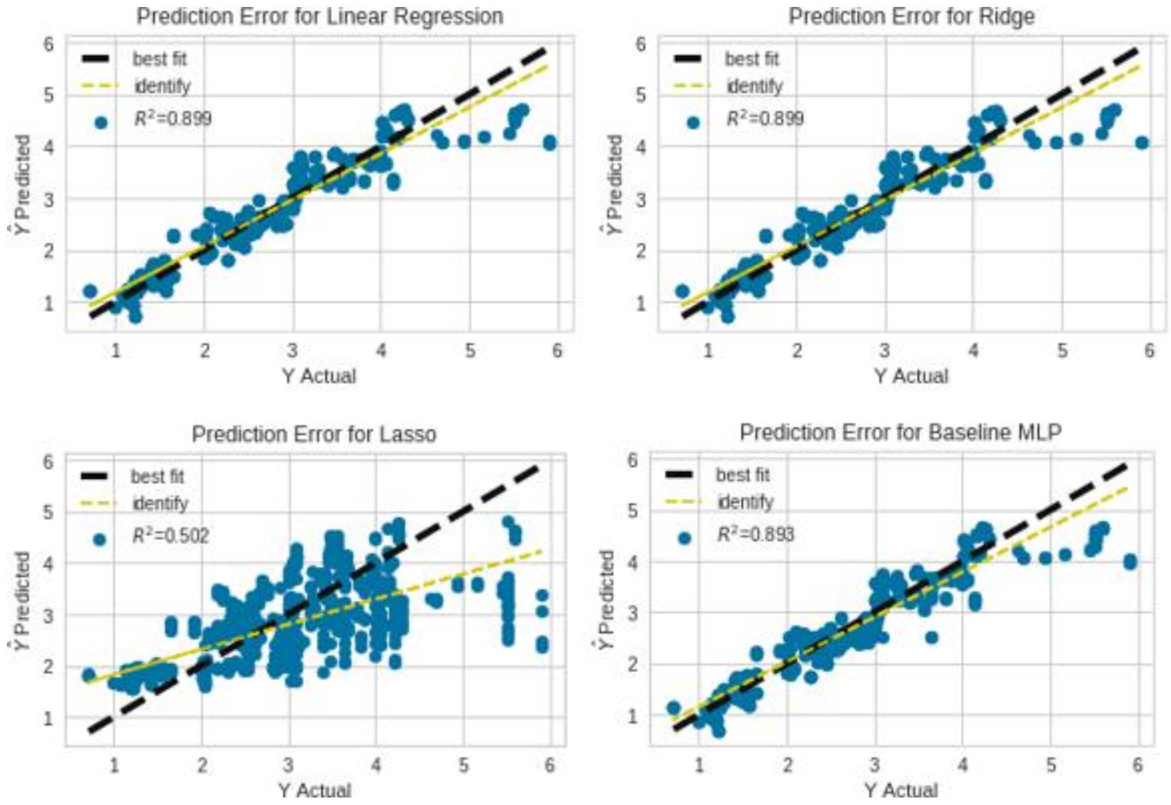


Figure 21. The baseline prediction errors and R^2 measures

The data was separated into training and test sets as 70:30 respectively of the total data sample. The predicted error was the difference between the prediction and the actual observed value and is defined as the following:

$$e_{T+h} = (y_{T+h} - \widehat{y_{T+h|T,}}) \pm 10\% \tag{21}$$

Where $Y_1, \dots, Y_t$ is the training data and $Y_t + 1, Y_t + 2, \dots$ is the test data and 10% error margin. The baseline MLP model was implemented using 4 nodes in the input layer, a single fully connected hidden layer with the linear rectifier activation function and finally for the output the ADAM optimisation and mean squared error loss was applied. Once a baseline was established it was possible to explore further ways to improve the performance by creating deeper more complex networks, which included, Random Forests (RF), Convolutional Neural Network (CNN) and gradient boosted decision trees (XGBoost). The prediction error and the R^2 measures of the three deep neural networks and the boosted tree model are shown presented in ***Figure 22***.

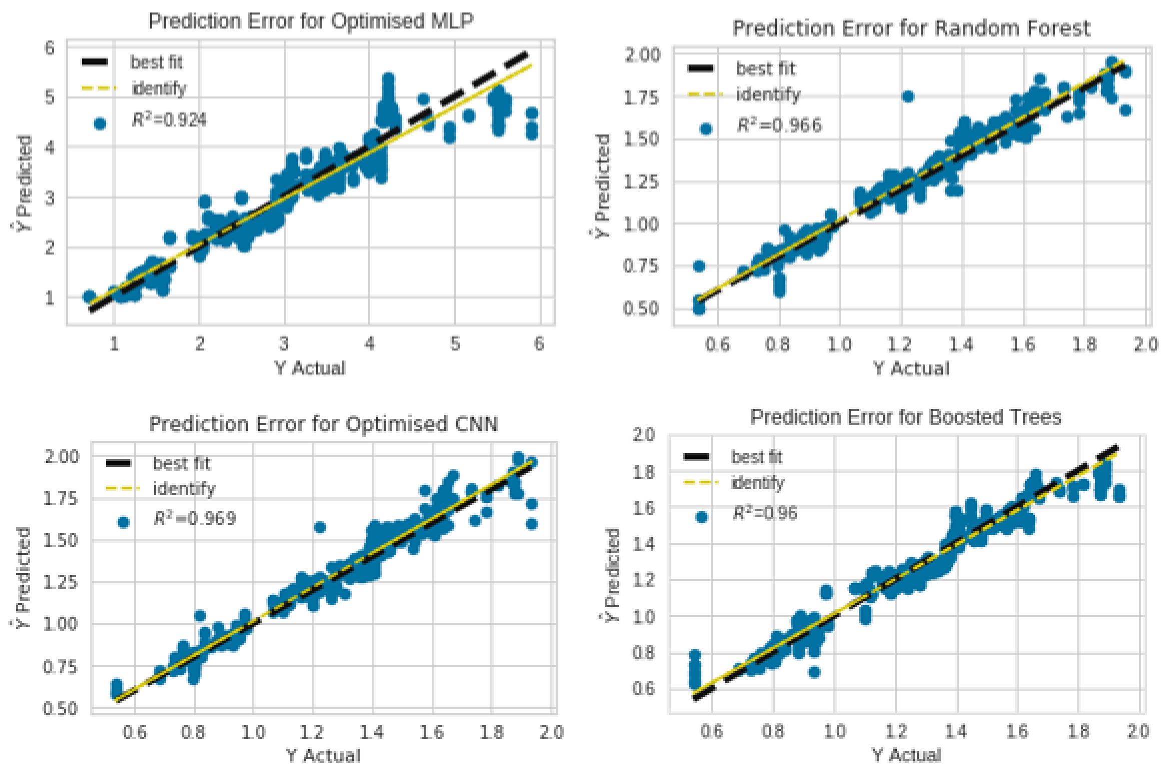


Figure 22. The optimised model prediction errors and R^2 measures

The performance of all experiments was evaluated by repeated 5-fold cross-validation. The experiments were repeated 10 times, and their mean scores are presented in ***Table 9***. The table shows the mean (%) accuracy scores for each model in the experiment. By eliminating correlated features that degrade the performance and systematically selecting the optimal feature subset enables models to perform better.

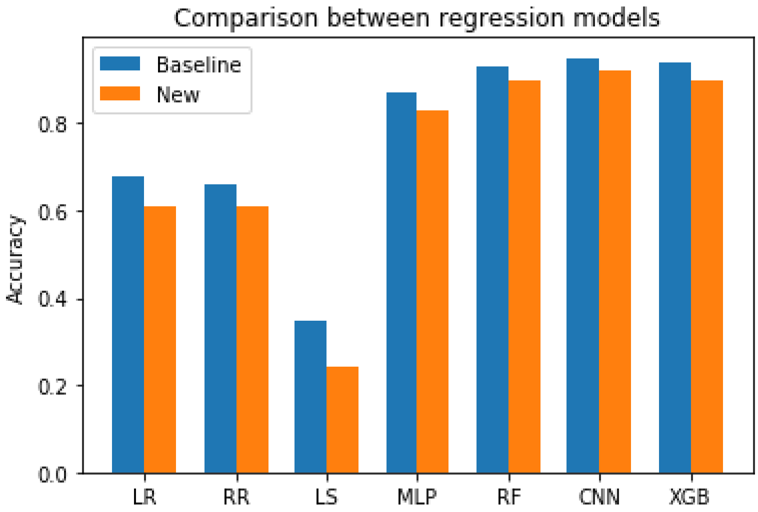


Figure 23. Comparison of proposed approach vs the baseline model across all models

Table 9. MET Predictive mean (%) accuracy scores

Model	Baseline Accuracy	New Approach (time seconds)	Accuracy Diff
Linear Regression	*68.0% (+/- 6.20%) (10.9s)*	*61.7% (+/- 6.80%) (6.5s)*	*~10%*
Ridge Regression	*66.2% (+/- 5.02%) (11.8s)*	*61.7% (+/- 7.02%) (7.0s)*	*~8%*
Lasso Regression	*31.1% (+/- 8.07%) (12.14)*	*24.1% (+/- 9.00%) (7.4s)*	*~23%*
MLP	*87.2% (+/- 4.07%) (90.46s)*	*83.5% (+/- 6.07%) (55.5s)*	*~5%*
Random Forest	*93.0% (+/- 3.01%) (228.03s)*	*90.2% (+/- 5.50%) (138.2s)*	*~4%*
CNN	*95.1% (+/- 4.01%) (434.11s)*	*92.6% (+/- 6.10%) (285.6s)*	*~4%*
Boosted Trees	*94.0% (+/- 3.42%) (37.10s)*	*90.1% (+/- 5.44%) (21.2s)*	*~5%*

As shown in ***Figure 23***, LR and RR achieved accuracies within 10% of the baseline. LS achieved accuracy within 23% of the baseline. The remaining models MLP, RF, CNN and XGB all achieved an accuracy within 5% of the baseline all also improving on the computational time. The initial threshold was set to 0.5 and increased until the models achieved scores within 5% of the baseline. The final threshold value was 0.8.

When the PCA is applied the output values corresponding to the different activities ('Supine', 'Lego', 'WalkingSlow', 'WalkingMed', 'Running', 'ThrowandCatch', 'Kicking', 'Cycling' etc) is projected onto a lower-dimensional feature space. Such the reduction in the model size reduces the computational load and hence reduces the computational time as shown in ***Figure 24***.

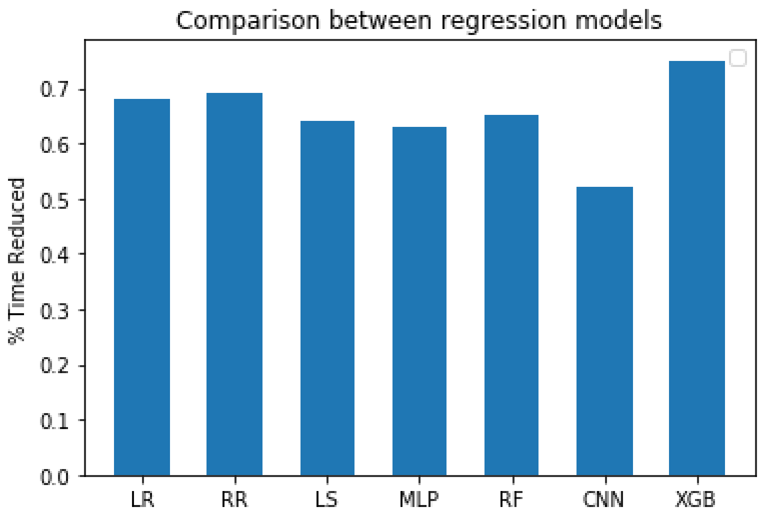


Figure 24. Percentage reduced in computational time

The proposed approach combining PCA and VIF can maintain the performance of the ML model while significantly reducing the computation time. Applying the PCA immediately after the feature elimination is completed, substantially reduces the model size. The results show that the approach does speed up the computational time. The computational time was reduced by more than 50% for all the models.

5.3 Discussion

A lightweight feature selection approach for ML developed in this research has proven to be useful in deploying to a low powered device (Rasberry Pi3). This case study extends the understanding related to accelerometer assessment of children's physical activity utilising the lightweight ML approach. This study is first to evaluate appropriate lightweight ML models to accurately approximate the directly measured energy cost of physical activity in children. This means it is now possible to more effectively measure PA without focusing too much attention on the limit resources and latency issue concerning the need for cloud-based services.

A key strength of this study is using lightweight ML techniques as ML can recognise patterns in an acceleration signal rather than merely using the magnitude of acceleration for prediction. The present study results support the use of deep learning techniques as viable approaches to analysing accelerometer derived movement data in children. The MLP model produced excellent results (above 80%) after adding a 4 node fully connected hidden layer with a relu activation function which was modelled over 1000 epochs. Finally, the results of the random forests, convolutional neural network and gradient boosting machine models were able to attain the highest levels of accuracy with a mean average score of above 90% when predicting the energy expenditure (METs) for physical activity using the data produced by the wrist-worn accelerometer. The present study results are congruent with prior work undertaken in adults by Montoye, Westgate, et al. (2018), which reported that machine learning models predicted physical activities with accuracy between 71-92% from wrist-worn accelerometers. Finally, the XGBoost model was implemented as it achieved over 90% accuracy consistently and also reduced the computational time by 75% from the baseline time.

The results of the present study provide a robust foundation for further work refining the utility of ML approaches to classify physical activity in children better. There are, however, some limitations of the current study. We acknowledge that the data presented here are based on activities undertaken in a laboratory setting. The activities selected were representative of those undertaken by children for physical activity and included locomotor activity, cycling and object control skills. However, additional research is needed, which replicates the current work using a more comprehensive range of activities as well as examining the utility of lightweight ML approaches to classify physical activity undertaken in free-living environments. Such work will be useful in further training the lightweight ML models and increase the accuracy of prediction on energy cost and METs. The research presented in the current study also has a practical application. With the increasing prevalence of self-monitoring of physical activity behaviours using wearable technology accurate estimation of energy, expenditure is critical to use of accelerometry or wearable technology for large scale physical activity monitoring or use as behaviour change tools. It would, therefore, be interesting to explore whether the accurate prediction of the energy cost of physical activities may encourage more physical activity and healthy lifestyle in children in the longer term (Ellis & Piwek, 2018).

5.4 Conclusion

The evidence from the present study shows that the novel lightweight feature selection approach to ML used to model PA data, can be used to accurately predict energy cost, METs, with 90% accuracy. Given the importance of physical activity for health benefit and emphasis on assessment of physical activity for population monitoring and accurate targeting of public health-related interventions, the refinement of physical activity measurement is vital. It is mainly the case for children where typical activity patterns are more sporadic and omnidirectional. The experimental study using the lightweight ML techniques showed that the energy expenditure from the accelerometer readings had a preference towards the dominant wrist or ankle as the movement in those positions were more consistent during the activity and marked those features in relation as more important. However, the algorithms did not show any significant gains in performance. The convolutional neural network performed slightly better than the random forest and gradient boosted machine; however, all three performed consistently high. Yet, the XGBoost model was selected as it achieved the best score while also reducing the average computational time by up to 75%.

Chapter 6: Case Study Two - Swim Classification and Tracking on a Low Powered Device

Swimming is one of the few sports that people of all ages and abilities can take part in, that involves the rhythmic movement of many muscle groups for an all-over workout (Mooney et al., 2017). Recent advances in wearable technologies have paved the way for better tracking of human Physical Activity (PA) in general. Moreover, with the advent of waterproof wearable devices, such as the Samsung Gear Fit, Apple Watches, FitBit and Misfit Shine, there has been an increase in pervasive PA tracking in swimming (Kos & Umek, 2019; Mooney et al., 2015). By monitoring one's physical activity while swimming, it can allow one to benefit from being able to measure their performance over time as a motivational tool (Mooney et al., 2017). The advances in kinematic swim sensor technology have allowed researchers to analyse physical stroke mechanics, swim performance and evaluation of exercise intensities (Mooney et al., 2015). Mooney et al. (2015) also indicate that technology also offers significant advantages over traditional computer vision-based approaches. The authors also suggest an essential consideration for ongoing feature extraction development work is still necessary. Therefore, the primary aim of this experimental study was to analyse the swimming activity signal data and explore appropriate machine learning approaches to accurately classify the stroke style, within a lap, by considering the temporal sequences. Secondly, to deploy a lightweight ML model that can run using the limited resources on the Gear Fit wearable low powered device (LPD).

This study has the following two contributions:

- Deployed a lightweight ML model on the LPD
- Using the proposed lightweight ML model accurately classify a swim style from the following activities; breaststroke, backstroke, butterfly, and freestyle.

6.1 Methods

6.1.1 Participants

The data was supplied by the School of Computing, Electronics and Mathematics at Coventry University. Adult swimming athletes who were proficient swimmers (218 subjects) in four swimming stroke styles, (freestyle, backstroke, breaststroke and butterfly), were chosen and grouped into three levels; beginner (7 subjects), intermediate (151 subjects) and advanced (59 subjects). The process of gathering and analysing the information was explained to all participants. No personal details were recorded or required.

6.1.2 Procedures

Each participant was required to swim continuously for 4 in each stroke style in a 25-meter pool. The device was worn on the left arm and was configured to record at 100hz at 1-second epochs to achieve consistency. Some participants also performed a further 4 laps of the medley (mixed stroke laps of 4). Between each of the 4 laps, there was a 1-2-minute rest period.

6.1.3 Data processing for Swim on the low powered device

After each participant completed their session, the tri-axial accelerometer, triaxial gyroscope, tri-axial linear accelerations and pressure readings are extracted on the LPD, as first-order primitives, for analysis. An excerpt of the raw swim sensor readings is shown in ***Figure 25*** directly from the terminal of the LPD. After initial observations of the raw data plots, several noise suppression algorithms are used to help smooth out the signal without losing essential details while also being efficient. The signal data from another participant that is manually labelled, as shown in ***Figure 26*** *and* ***Figure 27***, that corresponds to the change in activity during the swim are preloaded on the LPD to support the supervised learning models. The swimmer would begin swimming, and there would usually be a brief GLIDE period, as they kick off the wall, then a SWIM period, of a stroke style, then toward the end of the lap a FLIP_TURN and GLIDE phase for the returning lap.

timestamp	ax	ay	az	gx	gy	gz	lx	ly	lz	pressure
1481722951685226	0.600598	6.532406	7.042077	-96.25	-10.71	9.52	-0.358516	0.190475	-0.376171	1030.186523
1481722951692684	0.387637	6.45105	6.843473	-76.580002	24.85	29.33	-0.358516	0.190475	-0.376171	1030.186523
1481722951706559	0.579063	6.312266	6.836295	5.67	-3.01	-0.35	-0.224351	-0.205241	-0.332155	1030.186523
1481722951712721	0.689133	6.419943	6.898508	8.12	-2.45	-1.61	-0.028773	-0.204055	-0.467025	1030.186523
1481722951725297	0.753739	6.54437	6.908078	8.19	-1.89	-1.47	0.078739	-0.118339	-0.384944	1030.186523
1481722951732776	0.954736	6.750153	6.970293	6.02	-0.35	0.49	0.14126	-0.006139	-0.364204	1030.186523
1481722951745252	0.90688	7.008578	6.929615	2.87	0.21	3.01	0.336692	0.175374	-0.279582	1030.186523
1481722951752888	0.399601	6.07777	7.190433	0.07	1.05	1.4	0.282145	0.425106	-0.311793	1030.186523
1481722951765836	0.540778	7.771888	7.226324	-0.49	0.91	1.33	-0.235401	-0.513283	-0.043182	1030.186523
1481722951772893	1.18684	7.63789	5.319244	-1.26	3.57	2.38	-0.090186	1.189517	-0.015545	1030.186523

Figure 25. Raw Swim Sensor Reading

title	sample_start	sample_end	timestamp_start	timestamp_end	seconds_start	seconds_end
START	1	1980	1481722951685226	1481722971488296	11.25	31.06
GLIDE	1980	2282	1481722971488296	1481722974510597	31.06	34.08
FREESTYLE	2282	3738	1481722974510597	1481722989080336	34.08	48.66
FLIP_TURN	3738	3885	1481722989080336	1481722990552714	48.66	50.13
GLIDE	3885	4187	1481722990552714	1481722993575028	50.13	53.15
FREESTYLE	4187	5902	1481722993575028	1481723010735410	53.15	70.31
FLIP_TURN	5902	6019	1481723010735410	1481723011907558	70.31	71.48
GLIDE	6019	6311	1481723011907558	1481723014829151	71.48	74.41
FREESTYLE	6311	7948	1481723014829151	1481723031208908	74.41	90.79
FLIP_TURN	7948	8085	1481723031208908	1481723032581850	90.79	92.16
GLIDE	8085	8398	1481723032581850	1481723035712272	92.16	95.29
FREESTYLE	8398	10148	1481723035712272	1481723053236323	95.29	112.81
REST	10148	13319	1481723053236323	1481723084966346	112.81	144.54

Figure 26. Signal label markers of change of activity

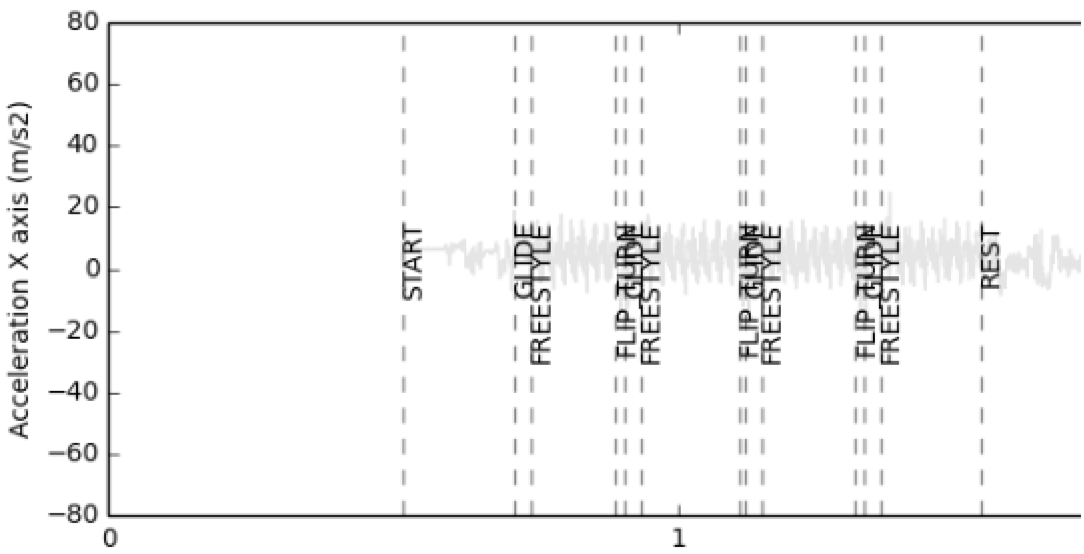


Figure 27. Signal plot with the corresponding label markers

The second-order primitives (mean, median and standard deviations) are derived from each first order features. The non-parametric analysis of the continuous one-dimensional probability test, also known as the Kolmogorov–Smirnov test, is carried out to compare the probability distribution for a good test fit. The next few sections describe some of the experimental analysis work that were carried out to help improve the classification accuracy of swim Activity Recognition (AR).

6.1.4 Manual Classification Method

The first working version of the algorithm was based on a manually tuned classification approach. The task to accurately classify the swim actions were further subdivided into two sub-tasks:

1. Determine the swimming style
2. Determine the beginning and the end of the lap

The second task is directly dependent on the first. By determining the swimming style, the algorithm recognises the swimmer is engaged in swimming and can, therefore, determine the start and end of the lap. By analysing the filtered signal, as shown in ***Figure 28***, the following observations were made:

- During the swimming phase the signal could be modelled as a constant function, plus the oscillations of the relative constant frequency, plus the error.
- Each style showed a different pattern of a relative mean value.
- During each turn the mean relative value was significantly affected.

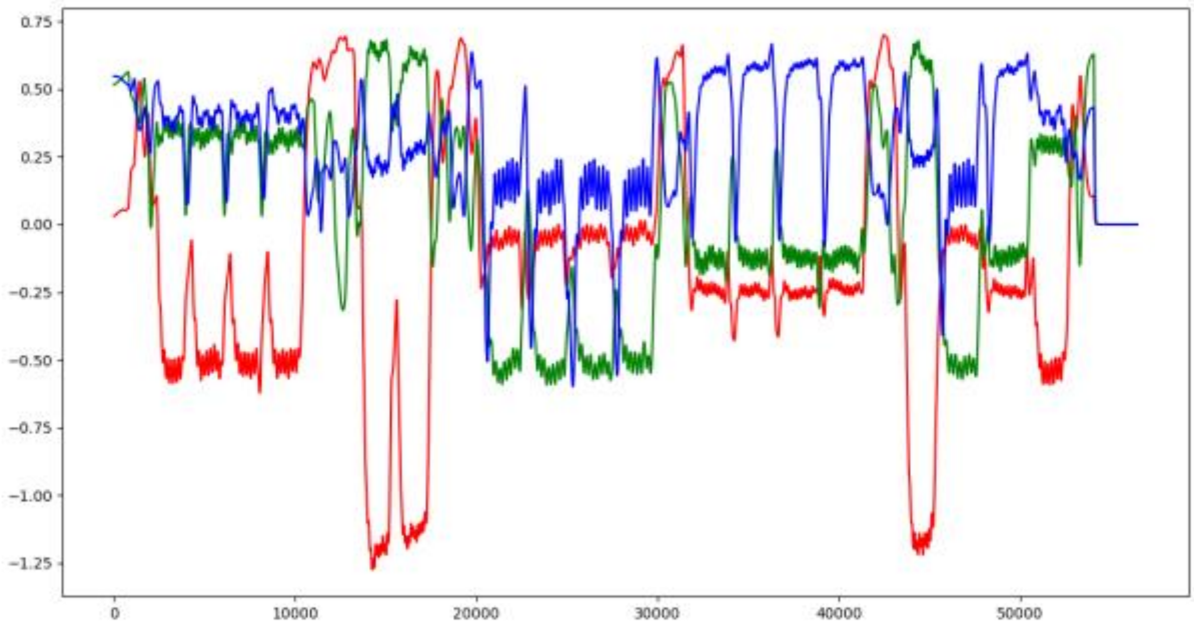


Figure 28. An accelerometer signal with a 3rd-order lowpass filter

The observations gave rise to the following assumptions:

- The value of the rate of change (i.e. derivative) can be used to detect turns.
- The relative values of the signal level can be used to determine the swimming style.
- The high-frequency signal (i.e. high pass filter) can be used to count strokes.

6.1.5 Low-pass Filtering

Formulating an algorithm to evaluate the low-frequency signal was a challenge. It must also fulfil the following two contradicting requirements:

- The filter must have a good roll-off, the high-frequency signal that is present must be cancelled out correctly and must be below the mean signal absolute value.
- The filter must also be relatively responsive; the low-frequency signal cannot be smoothed out too aggressively and not slow down the response rate.

The following statistical analysis tools were used to explore the signal:

- Moving window averages
- Exponential moving averages
- "Fast" moving average, modified in such a way that small amplitude in oscillations were dampened more than larger amplitude oscillations
- Classical lowpass filter

Exponential Moving Average (EMA) can be represented using the following formula:

$$EMA = [Close - EMA] * (2/n + 1) + previous\ EMA \tag{22}$$

If the factor equals 0, this algorithm always yields the initial value. If the factor is 1, it returns the unmodified signal. The smaller the factor is, the stronger the smoothing.
It is then possible to modify the algorithm by applying a nonlinear function to the deviation value.

The aim is to keep the value of the function to be as small as possible (as if the factor was close to 0) for fluctuations of small amplitude. Conversely, we want the value to asymptotically approach the linear function (as if the factor was close to 1) for fluctuations of large amplitude.
Various functions tried and tested until the following was eventually used:

$$min(abs(deviation) * threshold, 1) * deviation \tag{23}$$

The resulting modified moving average was good at cancelling the oscillations of small amplitude, and it reacted quickly for large deviations of the mean.

6.1.6 High-pass Filtering

The main problem with the high-pass filtering was that most of the noise lay in the high-frequency part of the spectrum. Therefore, amplifying the high frequencies yielded mostly noise.

The naive derivative output was noisy. The approximate derivative did not produce the expected result and could not be used. Applying the average to derivatives dampened the useful high-frequency signal and amplified the low-frequency signal; as a result, this yielded undesired results. The following exponential moving average algorithm was as used to smooth the signal:

$$\begin{aligned} &exponential\_moving\_average(signal \\ &\qquad - exponential\_moving\_average(signal)) \end{aligned} \tag{24}$$

The results of the exponential moving average filter applied to the signal is depicted in ***Figure 29***. It shows that filtered (red) signal using the modified exponential moving average (green) and the "fast" moving average (blue). The blue plot shows the desired dampening of the effect of the high frequencies without the losing low-frequency details.

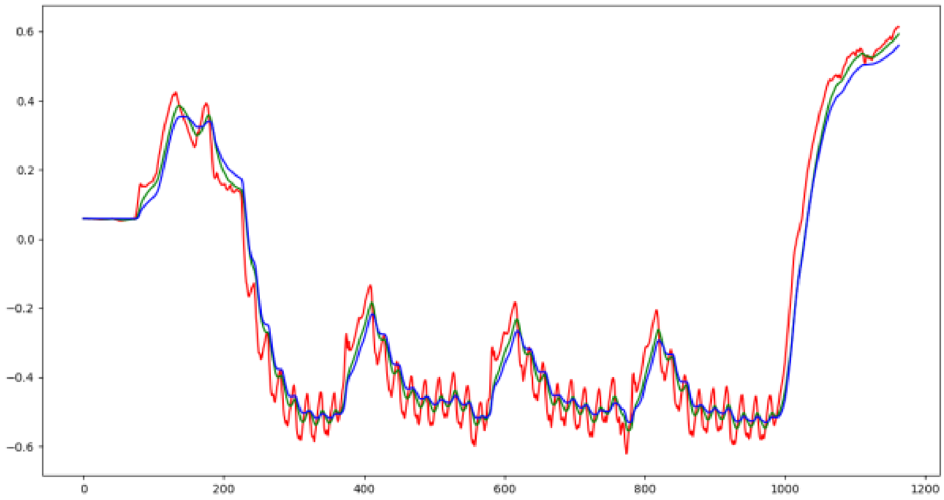


Figure 29. The result of the modified filter applied to the raw signal

6.1.5 The feature selection and reduction

Once signals from the raw sensors were smoothed out, the next step is the select the suitable features and reduce the model size using the novel lightweight feature selection approach.

The following steps were carried out:

1. Evaluate the threshold from 0.5 until the model achieves a score within 5% of the baseline.
2. Computation of the correlation matrix and check whether either condition is verified correlation analysis of the input features. None of the elements within the matrix was larger than 0.5.
3. RFE was used
4. RPCA is performed to further reduce the feature space dimension after eliminating collinear features, retaining 80% of the variance of the original dataset.

An additional experiment was performed to understand how the split of the initial dataset into training and testing parts is influencing the accuracy of the trained model. For example, splitting the data into training and test in consecutive chunks instead of randomly choosing the points may result in lower efficiency. The worst cases, using beginner-level swimmers, the accuracies of the different classifiers varied between 40% and 60%. In that case, 43% of the data were used for training and 57% for testing.

To better understand the dataset and with the aim to proceed with the recursive feature selection method, the correlation analysis of the different features in the feature space was performed. From ***Figure 30***, some of the features were correlated, such as the accelerations and positions along the same axis x, y and z (correlated terms: a_x and l_x, a_y and l_y, a_z and l_z); however, not significantly enough or above the initial threshold.

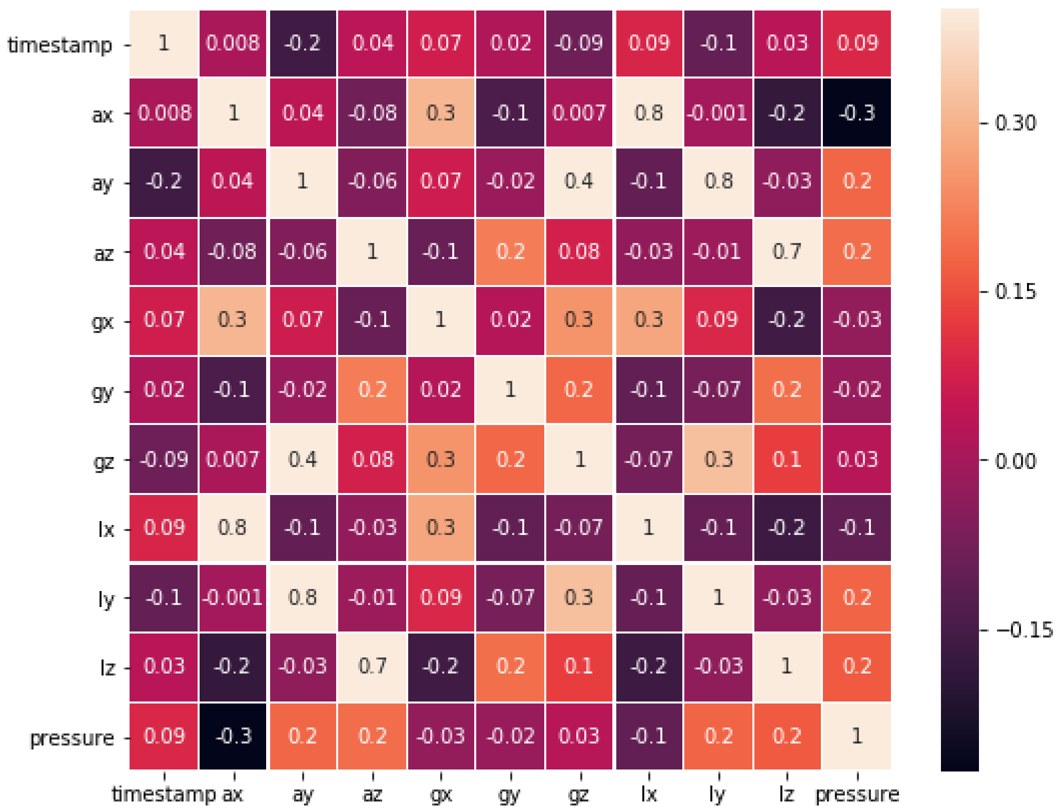


Figure 30. Correlation analysis for the swimming activity dataset

The second pathway of this approach of feature selection is based on the Variance Inflation Factor (VIF), a statistical indicator that can be calculated as the quotient of the variance in a model with multiple terms by the variance of a model with one term alone. The VIF represents an index that measures how much the variance of an estimated regression coefficient is increased due to collinearity. Features associated with high VIF can be removed from the input feature space (VIF > 0.5). This work will try to establish a synergy between the recursive feature estimation method and the application of the VIF. In the case of the swimming data set, VIFs are all lower than the lowest threshold of 0.5. Therefore, the recursive feature elimination was required. However, from ***Figure 31,*** VIFs of timestamp and pressure are significantly higher than others which may indicate their collinearity.

Finally, to point out that splitting of available data into training and testing data-sets may affect the amount reduction of the data-set through recursive elimination. Furthermore, classifier performances depend on the choice of training and testing data-sets. For example, the decision tree classifier works very well when testing data are extracted randomly from the available data. In contrast, its accuracy becomes significantly lower (in the order of 40%) when training set and test set include data belonging to disjointed time intervals.

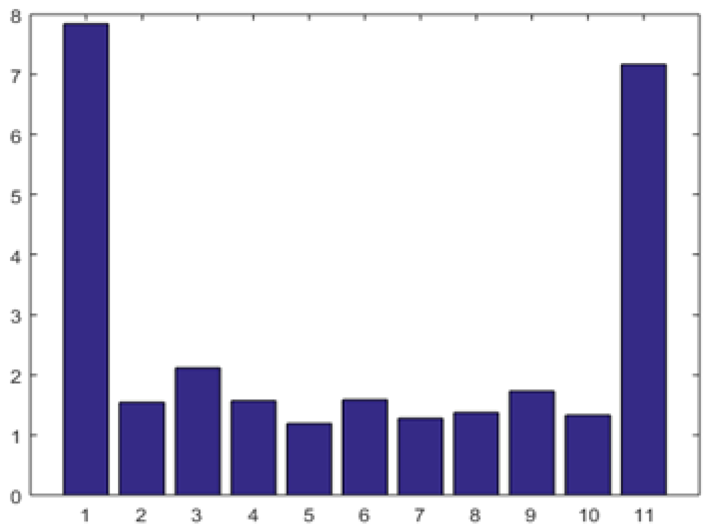


Figure 31. Variance Inflation Factors for the swimming activity dataset (1. timestamp, 2. ax, 3. ay, 4. az, 5. gx, 6. gy, 7. gz, 8. lx, 9. ly, 10. lz, 11. pressure)

6.1.7 Comparison of classification models with the baseline

The neural network is an algorithm that can be used for classification problems. It comes in many variants. For the solution, a lightweight MLP with 3-layer perceptron was chosen. The neural network is presented with an input vector of numbers. The network returns a one-bit output, based on the vector that belongs to a subset of the classified space. The network also has many outputs. In this case, it can independently determine whether the vector belongs to one or more subsets. Many independent single-output neural networks could obtain the same results. However, many-outputs can lead to an increase in the computational load.

The size of the first layer is equal to the size of the input vector. The size of the last layer is equal to the number of outputs. The size of the middle layer may vary. Each layer is connected to the previous one by synapses. If the sizes of the layers are A, B, C, then the number of synapses is:

$$\big((A+1)*B\big)+\big((B+1)*C\big) \tag{25}$$

Reducing the input layer any further caused the training never to converge. The larger the middle layer is, the faster the network converges in terms of the number of iterations, but each iteration takes longer. A compromise was made between reducing the number of iterations and the time of each iteration. The middle layer size of 50% to 75% of the input layer size was a good compromise.

The neural network accepts input in binary format. The solution takes inputs as a vector of numeric values 1 and -1. The input values were first quantized. A set of N thresholds is chosen, then a vector consisting of N bits are prepared. If the input value is greater than Nth threshold, the bit is set to 1, otherwise to -1. Increasing the threshold reduces the algorithm's efficiency while increasing accuracy. The thresholds values were set between 2 to 7 were tried. The thresholds can be distanced evenly, or they can be manually tuned.

The MLP models is given a pair of input and output vectors. The network will produce the output vector when given an input vector. The training process tries to minimize the cost function, which is $(x-y)2/2$, where x is the "current" output to be modified, and y is the desired output. When the error value was no longer diminishing significantly, and close to 0, the training was considered completed.

The network used in the algorithm had 6 outputs, each corresponding to one of the results: REST, TURN, BREASTSTROKE, BACKSTROKE, BUTTERFLY, FREESTYLE. Only one output neuron was set to 1. When more output neurons were active, or none, the output was considered invalid.

Based on the literature review the following conventional classification models were considered for the comparative study; Logistic Regression (LR), Decision Trees (DT), Lasso (LS) with the binomial argument, Multi-layer Perceptron (MLP), Random Forests (RF), Convolutional Neural Network (CNN) and Gradient Boosted Decision Trees (XGB). These classifiers have been preliminarily tested on a measured dataset representing the swimming activity. The splitting between training and testing datasets has in the first instance been obtained by extracting the testing datasets randomly. When the error value was no longer diminishing significantly, and close to 0, the training was considered completed. Furthermore, the PCA has been applied as a pre-processing step to get a reduced space set of features. All the classifiers achieve a more than satisfactory level of accuracy. When the PCA is applied the output values corresponding to the different activities (‘START’, 'GLIDE', 'FREESTYLE', 'FLIP_TURN', 'REST', 'BUTTERFLY', 'TOUCH_TURN', 'STOP') will be projected onto a lower-dimensional feature space. Such a reduction will make the feature space more crowded of output points.

From the results in ***Figure 32***, for the classification algorithm, it will become more challenging to determine decision boundaries accurately to separate the different decision regions, therefore, obtaining a lower classification accuracy when the PCA reduction is applied.

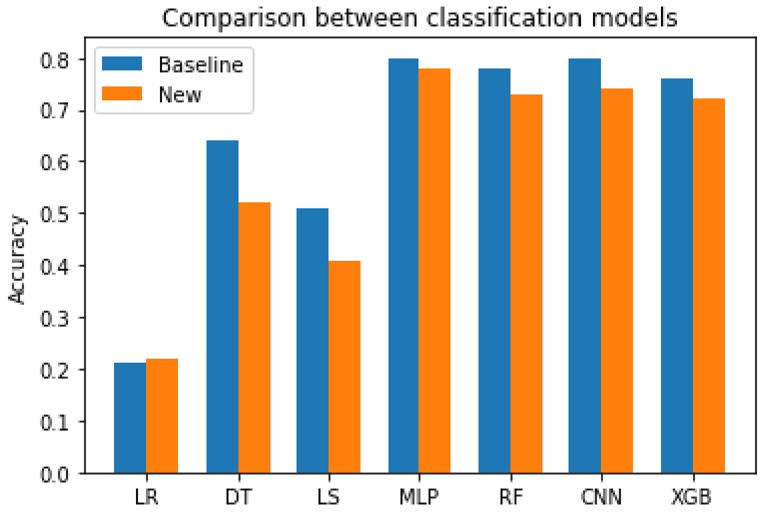


Figure 32. Comparison of classification models for the swimming activity classification

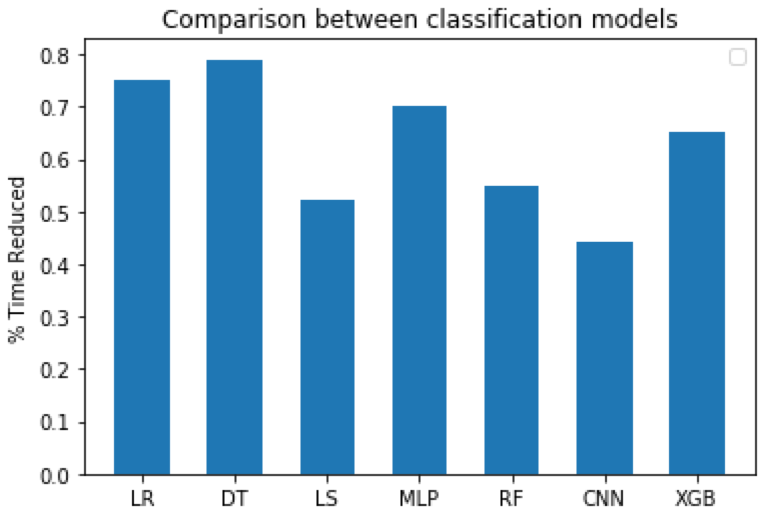


Figure 33. Efficiency gains for the swimming activity classification

In this case study, the results show that by combining PCA and RFE, as described in the proposed approach, it can maintain the performance of the ML model while significantly reducing the computation time across all the classification models shown in ***Figure 33***. Applying the PCA immediately after the feature elimination is completed, substantially reduces the model size. The results show that the approach does speed up the computational time reducing it by more than 45% for all the models.

The final model was tested on all available data, and the average scores for ten runs are shown in *Table 10*. *Figure 34* shows the final performance scores of the best performing model.

Table 10. Overall swimming activity classification scores

Model	Baseline % Accuracy (% variance) (time taken in seconds)	New % Accuracy (% variance) (time taken in seconds)	Accuracy % Diff
Logistic Regression	*21% (+/- 5.1%) (6.12s)*	*22% (+/- 7.30%) (3.5s)*	*~5%*
Decision Trees	*64% (+/- 5.50%) (3.94s)*	*52% (+/- 6.80%) (2.2s)*	*~19%*
Lasso	*51% (+/- 5.22%) (5.17s)*	*41% (+/- 6.20%) (3.4s)*	*~20%*
MLP	*80% (+/- 4.12%) (8.67s)*	*78% (+/- 5.01%) (5.1s)*	*~3%*
Random Forest	*78% (+/- 5.11%) (39.06s)*	*73% (+/- 6.01%) (25.2s)*	*~7%*
CNN	*80% (+/- 4.10%) (65.66s)*	*74% (+/- 5.01%) (45.6s)*	*~8%*
Boosted Trees	*76% (+/- 5.41%) (32.68s)*	*72% (+/- 6.12%) (19.2s)*	*~5%*

```
100% LAP ACCURACY : 88 / 120 : 73%
1 LAP OUT ACCURACY : 19 / 120 : 16%
2 LAPS OUT ACCURACY : 2 / 120 : 2%
3+ LAPS OUT ACCURACY : 11 / 120 : 9%

STROKE TYPE ACCURACY: 69 / 88 : 78%
STROKE TYPE 1 LAP OUT ACCURACY: 8 / 88 : 9%
STROKE TYPE 2 LAPS OUT ACCURACY: 5 / 88 : 6%
STROKE TYPE 3+ LAPS OUT ACCURACY: 6 / 88 : 7%

LAP TIME ACCURACY: 23 / 88 : 26%
LAP TIME 1 LAP OUT ACCURACY: 25 / 88 : 28%
LAP TIME 2 LAPS OUT ACCURACY: 15 / 88 : 17%
LAP TIME 3+ LAPS OUT ACCURACY: 25 / 88 : 28%
LAP TIME AGGREGATE ACCURACY: 883 / 1383 : 64%

STROKE COUNT 100% ACCURACY: 49 / 69 : 71%
STROKE COUNT 1 LAP OUT ACCURACY: 11 / 69 : 16%
STROKE COUNT 2 LAPS OUT ACCURACY: 4 / 69 : 6%
STROKE COUNT 3+ LAPS OUT ACCURACY: 5 / 69 : 7%
STROKE COUNT AGGREGATE ACCURACY: 994 / 1383 : 72%
Time taken: 77.53s.
```

Figure 34. Accuracy results from the final MLP Algorithm

6.2 Discussion

The training set ranged from up to three datasets. The size of the training set did not have an impact on the training time. The size of the middle layer roughly corresponded to the number of vectors the network could learn. A shallow neural network (MLP) with only 3 layers and 3 nodes in the middle layer could correctly classify a single dataset.

The algorithm did complete in less time, and the results were comparable to the baseline model. If two neurons in the middle layer detected the same activation pattern when running on the corresponding input vectors, then they are redundant. A periodic check was run if neurons in the middle layer detected correlations. If any did, the other neurons were removed, except one in each correlated cluster. Then new neurons could be reinserted. It allowed the model to achieve convergence more quickly. Some of the input points from within the input space do not uniquely belong to any of the outputs. Some inputs occur while the subject is turning. The network finds it challenging to identify what the swimmer is doing when presented with such a vector. Removing those vectors from the training set showed a great reduction of errors. The sample space is divided into rectangular regions, as shown in ***Figure 35.*** The curve in the middle is a boundary between two outputs. Green rectangles correspond to one group of outputs, red to other outputs. The blue rectangles are ambiguous and must be removed from the training set.

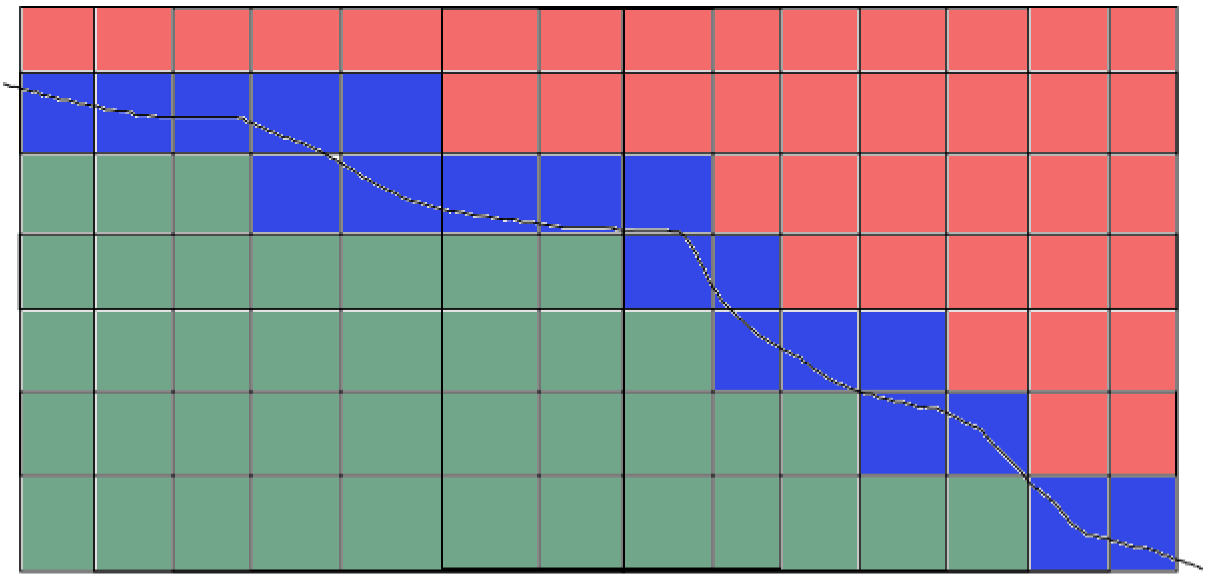


Figure 35. The sample space is divided into rectangular regions

For each output, the training set contains "positive" vectors (when the input should produce certain output) and "negative" vectors (when the input should not produce it). If the training set contains too much of one kind of vectors, it will often default to the trivial solution: either always show positive or always show negative. Balancing the number of positive and negative vectors for each output helped to solve that problem. This was especially the case with turns, which happens very rarely and are very short, so they are under-represented when the training set is not balanced.

For each output, some input vectors occur more often in each subject. Measurements showed that 10% of the most common vectors covered 50% data points and 50% of the most common vectors covered 95% of data points. Knowing that neural networks have a limited capacity, it was better to limit the training set to only the most common vectors. Measurements showed that the network could correctly classify about 300 input vectors. This optimization led to a greater improvement of the classification accuracy.

After each iteration, it was possible to checked and verify how much the network had learnt from the vector, and if there were no gains in performance, it was trained again. In the more advanced set up a list of vectors were created and ordered from most frequent to least frequent. The network was checked if it can perform well in classifying them. The most frequent vector that the network could not classify was then presented. This optimization helped to prevent the network from "forgetting" what it had learned before.

When a network has been trained, it can be checked for errors. The errors can be false positives (signal occurring where it should not) or misses (signal not occurring where it should). However, when the output label was TURN this approach was unable to perform as well as the other labels. The moving time window approach performed considerably better for the remaining output types and above 70% accuracy. The error state was defined as follows:

1. If at any time point there is a non-swim signal and there is no turn phase +-3s from that time point, it is a false positive.
2. If at any time point there is a turn phase and there is no non-swim signal +-3s from that time point, it is a miss.

While this procedure did help to minimise non-swim errors, it also reduces the accuracy of stroke recognition. Given the requirements, this trade-off was deemed acceptable. The network's convergence optimisation had greatly improved and the training time and reduced it to about 20 minutes.

One of the most important factors was the choice of training subjects. Some subjects contained errors and were discarded from the training. Some caused a drop in accuracy when included in the training set. It is believed that the most "average" swimmers should be included in the set and "outliers" should be avoided to maintain the accuracy levels for the widest group of swimmers.

6.2.1 Improving turn detection and timings

The neural network described in the previous chapter worked well for stroke-type recognition; however, it performed poorly with turn detection. Trying to improve turn detection reduced the accuracy of stroke-type classification. The output of the neural network required further post-processing (including passing through another neural network).

The quality of filtering was crucial. As the network analysed the signal level, any unfiltered oscillations related to stroking caused the level to fluctuate and as a result reduced the predictor performance

.

The following improvements were made to help maintain classification accuracy:

1. To utilise low-pass filtering.
2. As the neural network performed well in detecting stroke type, further processing procedures were removed to maintain efficiency

The new turn detection algorithm was derived from the observation that no signal could be reliably assigned to turning. When the subject is turning, the signal doesn't produce a consistent pattern. However, the signals observed after the turn do show a consistent pattern. Detection of such a signal, together with its mean delay from the expected turn, could be used to establish turn timings. As the algorithm relies on recognisable signals being produced after the non-swim, it is necessary to introduce a delay of up to 8 seconds before reporting the lap.

6.2.2 Filtering and Vector Preparation

The low-pass filtering algorithm used earlier were based on the principle of weighted averages. The only difference was the choice of weights. In an attempt to improve the lap timings, calculating the median instead of the mean was tried. After the initial low-pass filtering (using the usual algorithm) the signal was probed every 0.1 seconds. It was then added to a moving time window of the length of 4 seconds or 40 data points. The length of 4 seconds was based on observations of the mean stroke time which was approximately 2.6 seconds. The 4 seconds allowed just enough overlap to compare the two mean strokes and identify the window's current activity. After every probe, the median of the window was calculated. The median did produce better classification stability over the mean value; it was responding more efficiently to workout type activity changes in sequence, such as swim > turn (non-swim) > swim. The stroke signal oscillations were also being accurately identified. The median was then used in place of the low-pass filtering algorithm. Calculating the median requires sorting the element and then finding the centre element, which is computationally expensive. Sorting takes n log n operations where n is the length of the time window. Even though the number of operations on a 40-points window would still be relatively small, a need for optimization arose considering limited CPU resources. Instead of sorting the window each time, two copies of the window were held in memory: unsorted and sorted values. When a new value was inserted into the window, the oldest value is removed from the sorted array (if there are more elements of that value, only one is removed). Then the new value is inserted into the correct position into the array to maintain the sorted array. Only $n/2 + log\ n$ operations are required. With data structures such linked list, this number can be further reduced to $log\ n$; however, this attempt wasn't deemed necessary if the outputs were produced within a reasonable time, less than 30 seconds. In order to reduce the noise further, the mean of 4 middle elements was taken. After the window is sorted, it was also possible to find its maximum and minimum. In order to remove outliers and reduce noise, the mean of elements 4 to 8 (from the top or the bottom respectively) were taken.

The resulting 3 numbers (minimum, median, maximum) were called an envelope. The following observations were made:

1. The median was affected by the stroking signal. However, it remained close constant during a single workout phase, fluctuated drastically just before and immediately after turn activities. Also, each stroke style had a different median value.
2. The minimum and maximum were also close constant too. They remained close to the median when the subject had minimal arm activity during the turns and rest periods.

The values of the envelope were quantized and presented to the neural network. ***Figure 36*** depicts the processed signal stream as follows: Signal (red) and its envelope: median (blue), minimum and maximum (green) with outliers removed.

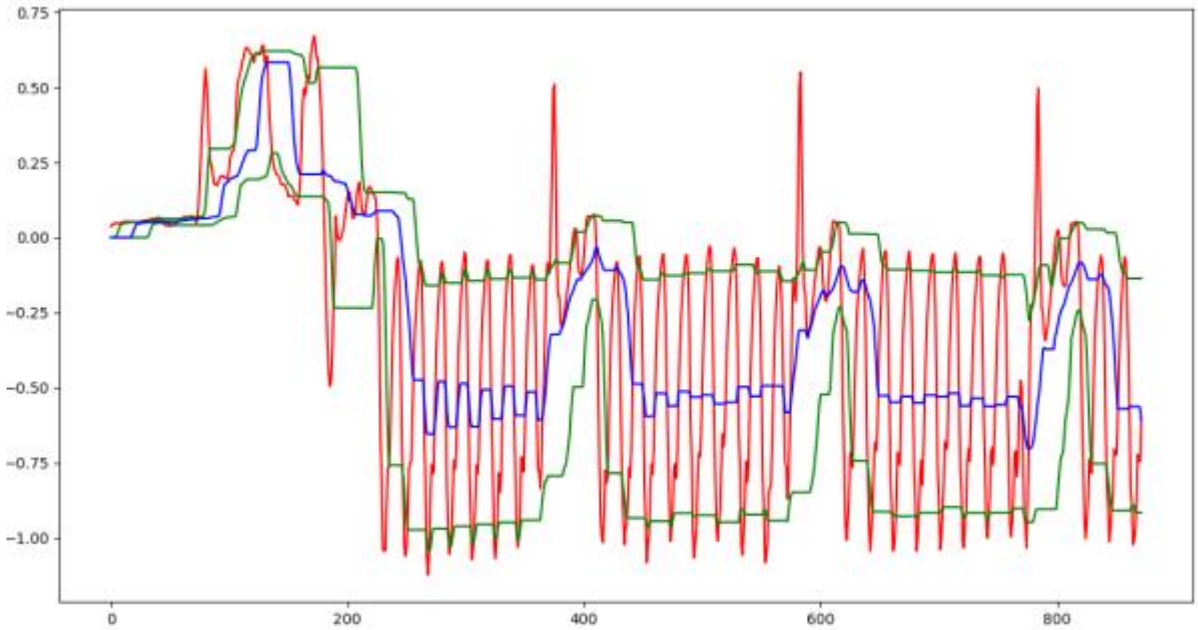


Figure 36. Sample from dataset 1; 4 second moving window.

6.2.3 Non-swim detection algorithm for lap counting

For each of the 9 data streams, the envelope (minimum, median, maximum) as defined in the previous chapter was processed the following way:

Four values were computed:

$$\begin{aligned} &1.\ median + minimum \\ &2.\ median - minimum \\ &3.\ median + maximum \\ &4.\ median - maximum \end{aligned} \tag{26}$$

The derivative, ratio of change, of each value was taken. The exponential moving average was applied to remove noise. The derivatives were then checked if they exceeded a certain threshold. If the derivatives exceeded the threshold, this was interpreted as a "non-swim signal". Observations showed that " non-swim signals" occurred reliably during non-swim, with a delayed trigger for confirming the non-swim of up to 16 seconds. Smoothing was applied to reduce "ringing". A moving time window was used to count how many times the derivative exceeded the threshold.

Only when a significant frequency of oscillations in the signal produced a derivative value above the mean threshold was the "non-swim " reported. Isolated spikes in the signal were therefore discarded. This also allowed the algorithm to report the beginning and the end of a non-swim. Each " non-swim signal" had an inconsistent ratio of change which was different from all other activities. In other words, the algorithm searched for this inconsistency to detect a non-swim activity. The next task was to measure the activity timings. 4 different times were taken:

1. time after the beginning of a lap when the non-swim signal is first detected
2. time after the beginning of a lap when the non-swim signal is last detected
3. time after the end of a lap when the non-swim signal is first detected
4. time after the end of a lap when the non-swim signal is last detected

There were 3 different non-swim signals, as shown in ***Figure 37***, detected (red, green, blue), occurring after non-swim (black spots, second row from the top, resting, turning and random activity).

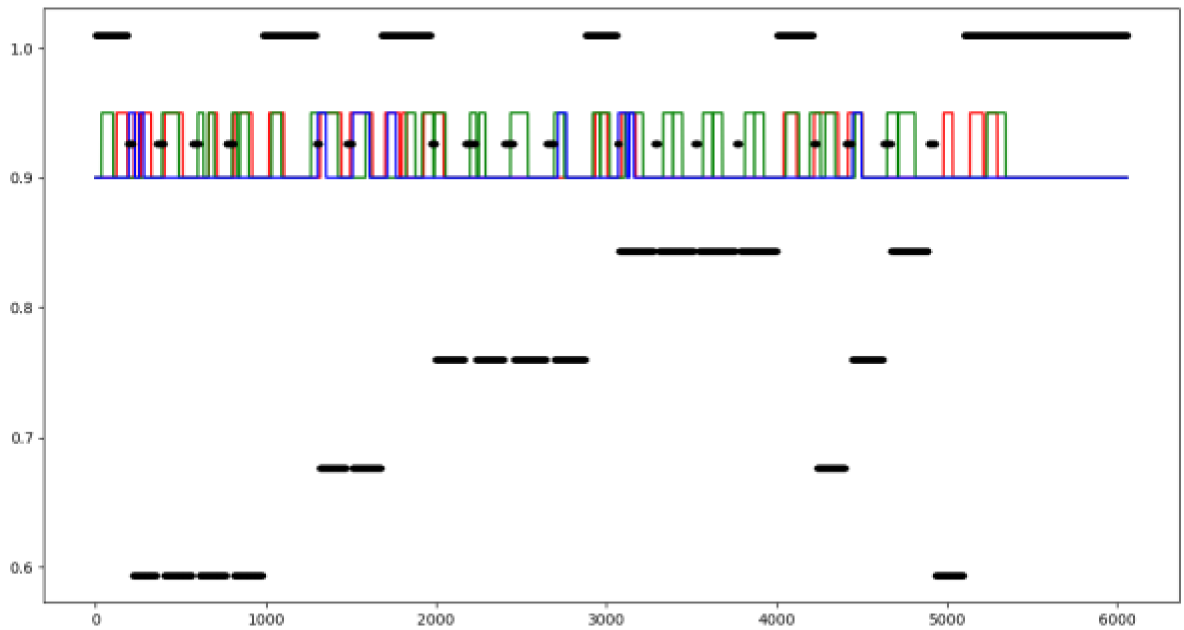


Figure 37. The detected "Non Swim" signals

The number of events and their times were recorded in a histogram. The length of the histogram was 8 seconds, the time resolution 0.1s, recording 80 data points. When an event occurred N tenths of seconds after the non-swim beginning or end as indicated by a truth table, the element N of the histogram was incremented. If the event happened more than 8 seconds after the last non-swim (beginning or end respectively), it was added to a special "other" counter.

To accommodate for limited time resolution, gaussian blur was applied to each histogram. The histograms mostly showed a normal distribution. Some showed two maxima, as they were a sum of 2 normal distributions. The mean was different for each non-swim signal, the variance good and within an acceptable range. The histograms were normalized so that their integral (including "other" counters) sum equated to 1. By examining the variance, the observation showed that the smaller the variance was, the greater was its maximal value.

The maximal value in the histogram can be understood as the algorithm's preference for the particular non-swim signal.

1. The more acute the variance, the more accurate were the timings
2. The more reliable the signal (occurring after non-swim, not happening far from non-swim), the greater the performance

With the histograms prepared, the algorithm was run the following way:

1. For each 9 data streams, prepare values in envelopes for each time point with resolution 0.1s.
2. Calculate derivatives of the functions described above. Test the derivatives for those that pass the mean thresholds. Filter out "ringing".
3. Take a moving time window of length 8s (80 points) with sets of numbers as elements. Disregard the last element, add an empty set as the first element.
4. Run the window through the proposed approach and detect the swimming activity within the 8-second window.
5. Whenever a non-swimming signal is suspected, record the start of the lap or the end of the lap
6. Take the sum of the elements within the set (start and end of the lap) and form a time series.

7. Run a peak detection algorithm on the time series to calculate strokes taken.

The peak detection algorithm, when run, as depicted in ***Figure 38***, can be identified in (red) using the sum of histograms for each non-swimming signal. The green spots are the detected non-swimming. The plot is from the data stream of a single swimmer.

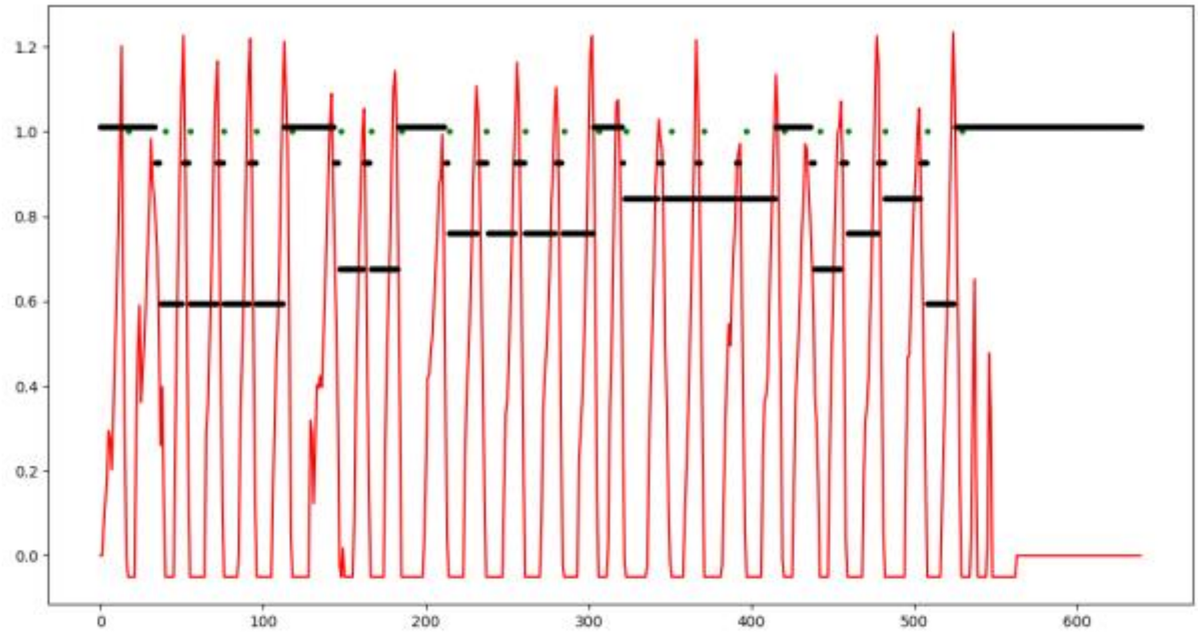


Figure 38. Non-swim signal detection

The resulting peaks correspond to non-swim with reasonable accuracy. However, if there is a non-swim signal detected, the exact timing was usually off by up to 3 seconds. This was due to detecting the inconsistent oscillations in the non-swim signal. The algorithm is typically unable to determine the non-swim signal when the oscillation amplitudes are less pronounced. The selected signal is generally taken when the amplitude exceeds the threshold or comes very close to it. If multiple signals were combined differently (average mean positions), the timings could be better. This could be a potential area for further research. After the beginning and the end of the lap was established using the algorithm above, the stroke type classification could be performed. A neural network without any post-processing was run on the envelope values. For each lap, the probability distribution was measured to determine the stroke style, and the highest probable stroke style was then reported. The output of the neural network was used to determine whether the subject is resting. The length of the 3d-vector consisting of median values of the linear acceleration was provided to the neural network. The data stream was reliable in determining the rest and swim periods. Only non-swim signals occurring inside the swimming phases were reported.

Stroke count:

To count strokes, the maximum value from the envelope of the accelerometer X-axis was taken. It was then subtracted from the initial low-pass filtered values of the X-axis. The resulting values were tested for the number of times it crossed the 0 level from negative to positive. The resulting number was returned as the number of strokes in each lap.

The algorithm achieved 73% accuracy in terms of lap detection, 78% in terms of stroke type classification and the stroke counts accurately counted at 71% of the time. The band-pass filter was used on the raw signal to extrapolate clear oscillations for the stroke count. Two low-pass filters with different cut-off were run on the raw signal; then their outputs were subtracted. The resulting signal was processed to find the number of times the signal crossed the 0 level. Based on the lap start end classifications the stroke counts were calculated only within this range, as shown in ***Figure 39***.

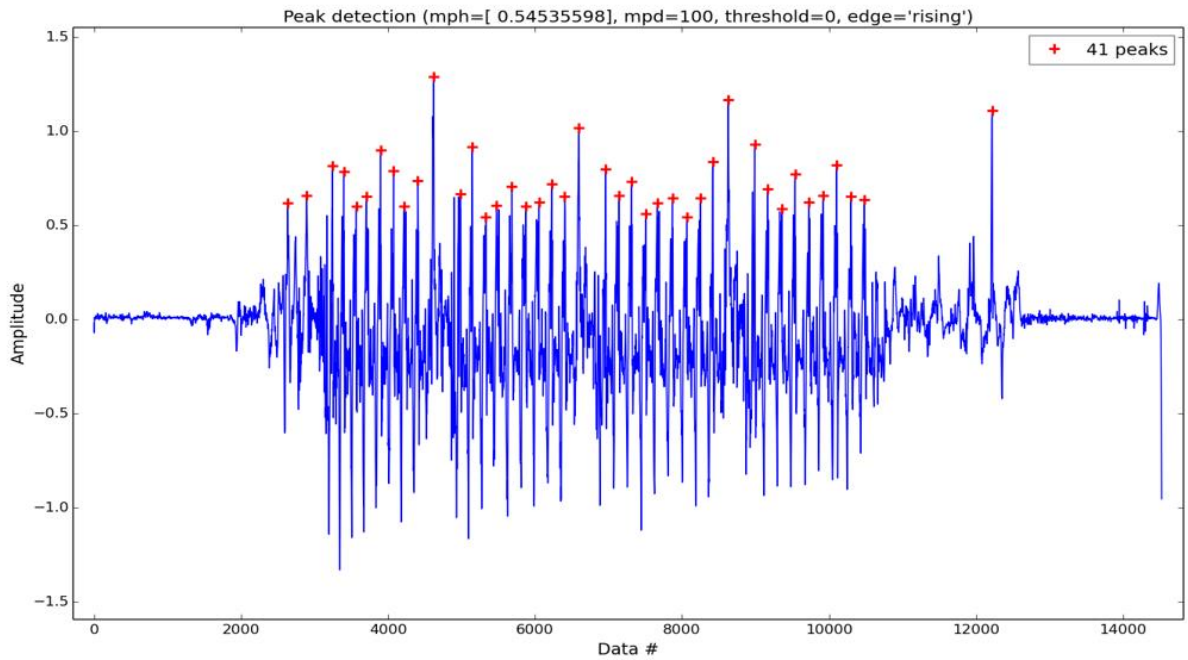


Figure 39. Stroke counting plot

6.2.4 Conclusions

This study aimed to evaluate the effective use of lightweight ML model to be deployed on wearable LPD and to detect physical activities in swim accurately. In this study, several signal classification models for activity recognition approaches were considered to identify swim activities. For most of the models, multi-labelled characteristics of data can create several challenges. They require sophisticated modelling and CPU-heavy methods such as kernel-based algorithms are more difficult to deploy due to constraints on the resources, especially with regards to the processing power of LPD. The proposed solution was able to reduce the number of features selected and significantly reduce the data model's dimensionality. The results showed that the proposed lightweight ML model using the classical neural network (MLP) could efficiently achieve above 85% accuracy and also be well within the 5% baseline model.

Chapter 7: Discussion

7.1 Findings Summary from the Case studies

When building machine learning models for Activity recognition, it is essential to understand the context of the activity within which it is being performed. Context-aware Computing is a broad field of research, and this thesis explored one of its many disciplines, namely activity recognition, with data gathered from IoT based devices and wearable sensors. In the following summary, the main findings and contributions are outlined.

This research aimed to deploy a suitable solution on low powered devices that can effectively operate within the limited computational resources available. The novelty of the approach is the combined use of the Variance Inflation Factor (VIF), the Recursive Feature Elimination (RFE) and the Randomised version of the Principle Component Analysis (RPCA) for feature reduction. VIF is computationally more efficient than RFE and this is because RFE includes a built-in modelling algorithm that runs iteratively whereas VIF is a single pass filter reduction. When there is low correlation between features, VIF detection of multicollinearity between features is not necessary. The RFE algorithm can achieve acceptable results in low feature correlation and reduce the number of features while it is computationally more demanding than VIF; it is less computationally challenging than RPCA on its own. The approach's objective is to reduce the number of features for improving data modelling efficiency while also being effective on low powered devices. Using RFE did not impact the data modelling efficiency but instead enhanced it by feeding it a reduced feature subset.

The initial experiment findings in **Chapter 4**, on the UCI public data sets, confirmed that the proposed approach is only practical on data sets that contain a large enough sample size with features with monotonic correlations that can be eliminated. The results showed that it could achieve similar reasonable performance within 5% of the baseline model in certain use cases while running on lower power devices. The work suggests that the proposed approach benefits situations that fit its specific criteria where there are correlated features. It has also proved to apply to both regression and classification problems. For binary classification, 45% efficiency gains were achievable, for multi-class classification, 50% efficiency gains were possible, and finally, 40% efficiency gains were potential for regression problems. However, the actual performance results indicate that effective practice will require significant improvements to enable this approach to be suitable for linear activity data problems.

The results from the first case study in **Chapter 5** show that the novel lightweight ML approach used to model PA data can predict energy cost, METs, accurately with 90% accuracy. A lightweight feature selection approach for ML developed in this research has proven to be useful in deploying to a lower power device (Rasberry Pi3). This case study extends the understanding related to accelerometer assessment of children's physical activity utilising the lightweight ML approach. This study is first to evaluate appropriate lightweight ML models to accurately approximate the directly measured energy cost of physical activity in children. This means it is now possible to more effectively measure PA without focusing too much attention on the limit resources and latency issue concerning the necessity of cloud-based services. This study's key strength is the use of the lightweight ML techniques as ML can recognise patterns in the time-series acceleration signal rather than merely using the magnitude of acceleration for prediction. The present study results support the use of deep learning techniques as viable approaches to analysing accelerometer derived movement data in children on the low powered device.

The experimental case study using the lightweight ML techniques showed that the accelerometer readings' energy expenditure had a preference towards the dominant wrist or ankle. The movement in those positions was more consistent during the activity and marked those features in relation as more critical. All the ML algorithms in the study showed significant gains in computational efficiency above 50%. The convolutional neural network performed slightly better than the random forest and gradient boosted machine; however, all three performed consistently high. Yet, the XGBoost model was selected as it achieved the best score while also reducing the average computational time by up to 75%.

The second case study in **Chapter 6**, evaluated the effective use of lightweight ML model deployed on wearable LPD and to detect physical activities in swimming activities accurately. In this study, several signal classification models for activity recognition approaches were considered to classify activities in swimming accurately. For most of the models, multi-labelled characteristics of data can create several challenges. They require sophisticated modelling and CPU-heavy methods such as kernel-based algorithms are more difficult to deploy due to constraints on the resources, especially with regards to the processing power of LPDs.

The size of the training set did not have an impact on the training time. The size of the middle layer roughly corresponded to the number of vectors the network could learn. A shallow neural network (MLP) with only three layers and three nodes in the middle layer could correctly classify a single dataset. The algorithm did complete in significantly less time, and the results were comparable to the baseline model. If two neurons in the middle layer detected the same activation pattern when running on the corresponding input vectors, then they are redundant. A periodic check was run if neurons in the middle layer detected correlations. If any did, the other neurons were removed, except one in each correlated cluster. Then new neurons could be reinserted. It allowed the model to achieve convergence more quickly. Some of the input points from within the input space do not uniquely belong to any of the outputs. Some inputs occur while the subject is turning. The network finds it challenging to identify what the swimmer is doing when presented with such a vector. Removing those vectors from the training set showed a significant reduction of errors.

The proposed solution reduced the number of features selected and significantly reduced the data model's dimensionality. The results showed that the proposed lightweight ML model using the classical neural network (MLP) could efficiently achieve above 75% accuracy and also be well within the 5% baseline model running on the LPD.

Thus far, the research findings provide consistent evidence of the effective use of the proposed novel approach performing well for linear time-series datasets. It is especially useful in analysing linear activity data generated by the tri-axial sensor readings. Therefore, it would be interesting to explore how well this method performs when applied to non-linear time-series activity problems.

7.2 Non-Linear Activity experiment

In addition to the linear activity experiments reported in chapter 4, 5 and 6, a further study has been carried out on how the proposed ML approach can be applied on non-linear activities. A spontaneous physical activity that does not produce a consistent signal and is omnidirectional, such as playing football and playing catch is considered a non-linear activity. A smart home experiment has been selected for this experiment.

Increasing research in activity recognition (AR) has become possible with the growing availability of low-powered wireless sensor networks that can continuously stream ambient data and enable researchers to draw inferences from (Mark Eastwood et al., 2019; Konios et al., 2018). Microcontrollers with sensors can monitor motion, power usage, light, temperature and door positions (open or closed) and stream this information to a central server which can record the streamed data in a suitable format. By analysing this data, it is possible to identify patterns in everyday activity a resident is engaged in and identify behavioural anomalies (Eastwood et al., 2019; Konios et al., 2018). With the growing need for elderly care as the demand has grown over the last few decades with the increasing elderly population (Lebherz et al., 2019; Onggo, 2012) this technology could support the social care system in improving the efficiency of care.

7.2.1 CASAS Smart Home

The Centre for Advanced Studies in Adaptive Systems (CASAS) provides a set of non-intrusive sensors developed to be easily integrated with a typical residential setting (Cook et al., 2013). The kit contains motion, door, light, temperature and power usage sensors (Eastwood et al., 2019). Eastwood et al. (2019) chose datasets number 24 (CASAS, 2015) for the experimental analysis of their approach. The data sample consisted of activity event readings for a single resident in an apartment. Normal behavioural patterns of Activities of Daily Living (ADLs) were learnt from the data collected from smart environments and used to provide well-being indicators and potentially recognising future health problems and identifying critical events such as falls or used to adapt the living environment in response to some health and safety concern raised through the analysis of the results (Eastwood et al., 2019). The data streamed from the smart homes' sensors are formatted as a sequence of a discrete set of events. Each sensor readings' events include the date and time, the sensor identification code, for non-binary sensors a Boolean value describing the sensor's status, and a trigger label was used to describe the start and end of activities as shown in *Table 11* and *Table 12*. The Activity Recognition or classification task can be described as a sequence classification task. By assigning an activity label to the sensor readings for each sequence of events (such as Sleeping, Grooming, Using the Toilet, Eating, Washing, Relaxing, Working, Reading). Eastwood et al. (2019) have found that Conditional Random Fields (CRFs) models are more commonly used in sequence labelling tasks and widely used in Activity Recognition as the model offers better discriminative classification performance by modelling the conditional distribution (Sutton, 2012; Tsochantaridis et al., 2005).

Table 11. Annotated CASAS raw data sample

Index	Sensor	Status	Trigger	Target	DateTime	Minute_of_Day
0	M021	ON	Sleep="begin"	Sleeping	2011-06-15 0:06:33	6
1	M021	OFF	NaN	Sleeping	2011-06-15 0:06:34	6
16	M021	ON	NaN	Sleeping	2011-06-15 3:37:47	217
17	M021	OFF	NaN	Sleeping	2011-06-15 3:37:48	217
18	M021	ON	NaN	Sleeping	2011-06-15 3:38:11	218

Table 12. Sensor ID and associated activity

Sensor ID	Activity	Simplified Activities
M021	*Sleeping*	*N/A*
MA013	*Grooming*	*(Personal Hygiene & Groom)*
MA009	*Toilet*	*N/A*
M005	*Eating*	*(Eat, Eat Breakfast, Eat Lunch & Eat Dinner)*
M007	*Washing*	*(Wash Dishes, Wash Breakfast Dishes, Wash Lunch Dishes & Wash Dinner Dishes)*
M004	*Relaxing*	*(Relaxing & Sleep out of Bed)*
MA014	*Working*	*(Work & Working at Table)*
MA010	*Reading*	*N/A*

7.2.2 Experiments

The CASAS organisations gather data over a few months from a smart apartment housing for a single resident and publish it on their website (Eastwood et al., 2019). The apartment labelled HH102 was fitted with a variety of different CASAS sensors that measure and report the temperature, light, motion, power usage, door usage, and the use of several tagged objects (Eastwood et al., 2019). The various sensor readings are recorded each time the state of the sensors changes. The readings included the Date, Time, Sensor Name, its Status, and a Trigger label to describe whether it's a start or end of an event, such as `Sleep="begin"` or `Sleep="end"`. There were 29 recorded activities consisted of 127 sensors trigger events (start or end). For a fair assessment of the new approach, as the experiment carried out by Eastwood et al., 2019, the same three derived features at a time t were extracted, the average of the sensor activation, the time since the sensor was last on/changed state and time of day as a value between 0 and 1. This produced a set of 135 features for further experimental modelling. The raw data was parsed to extract the features and sampled every minute with the corresponding label.

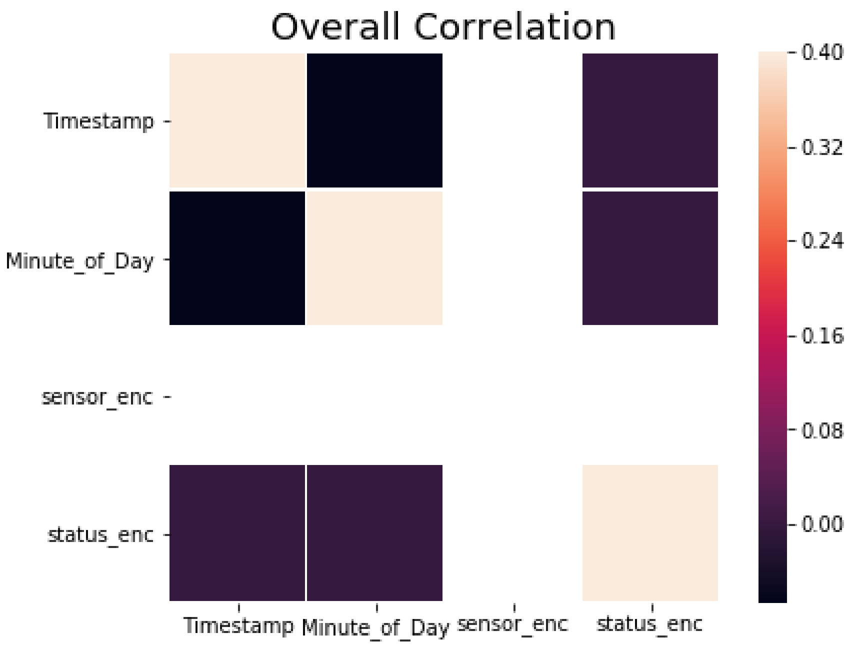


Figure 40. Correlation analysis for the input features of the smart home's activities

The correlation analysis shown in ***Figure 40*** highlights that some input features that are below the initial 0.5 thresholds. Correlation is very low between 'Timestamp' and 'Minute_of_Day' even though they both convey information about the time. Also, correlation is lacking between the feature 'status_enc' and 'Timestamp', 'Minute_of_Day'. In all the cases, the correlation between features is lower than 0.4. Due to the low dimensionality of the input space, these correlated features do not represent an obstacle to the identification of an accurate classification model even on a low computational power device. However, the aim is to evaluate if the classification accuracy of models can perform within 5% of the baseline model while also improving the computational efficiency.

The next step involved selecting the suitable features and reducing the model size using the novel lightweight feature selection approach. The data were randomly split 70:30 for training and testing, respectively.

The following steps were carried out:

1. Evaluate the threshold from 0.5 until the model achieves a score within 5% of the baseline.
2. Computation of the correlation matrix and check whether either condition is verified—correlation analysis of the input features. None of the elements within the matrix was larger than 0.5.
3. RFE was used
4. RPCA is performed to further reduce the feature space dimension after eliminating collinear features, retaining 80% of the variance of the original dataset.

The average scores of 10 runs are shown in ***Table 13.*** The results indicate the approach was significantly underperforming across all the models in the experiment.

Table 13. Overall binary classification activity scores for smart home's data

Model	Baseline Accuracy	New Approach (time seconds)	Accuracy Diff
Logistic Regression	*42% (+/- 2.2%) (2.01s)*	*25% (+/- 4.03%) (1.05s)*	*~40%*
Decision Trees	*55% (+/- 1.50%) (1.05s)*	*22% (+/- 3.22%) (0.8s)*	*~60%*
Lasso	*40% (+/- 1.03%) (1.26s)*	*28% (+/- 2.32%) (0.63s)*	*~30%*
MLP	*59% (+/- 1.30%) (3.78s)*	*49% (+/- 3.41%) (1.47s)*	*~17%*
Random Forest	*65% (+/- 1.60%) (2.66s)*	*49% (+/- 2.43%) (1.27s)*	*~25%*
CNN	*60% (+/- 1.10%) (4.29s)*	*39% (+/- 1.23%) (2.43s)*	*~35%*
Boosted Trees	*62% (+/- 1.20%) (2.21s)*	*37% (+/- 3.61%) (1.46s)*	*~40%*

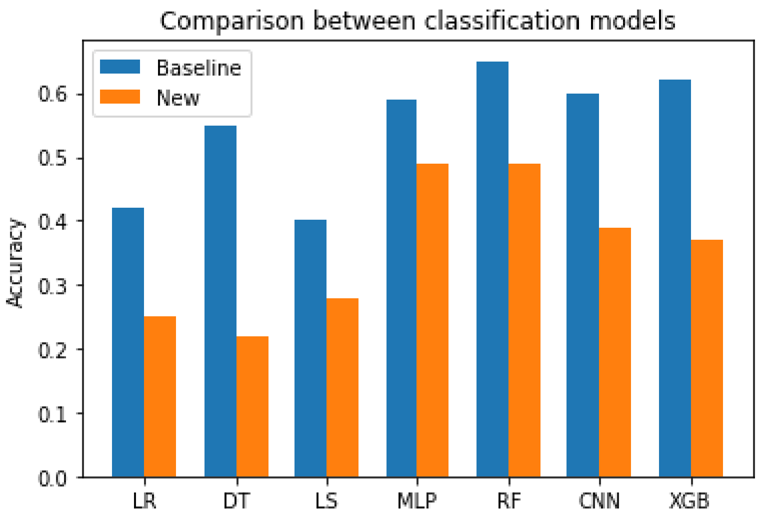


Figure 41. Comparison of binary classification scores for smart home's data

Figure 41 shows the final accuracy scores across the classification models between the baseline and the proposed approach. ***Figure 42*** shows the efficiency gains across all the classification models. The efficiency gains achieved are above 20% and up to 60%. However, the accuracies of the models were below 17% from the baseline and up 60%, which is significantly below the baseline.

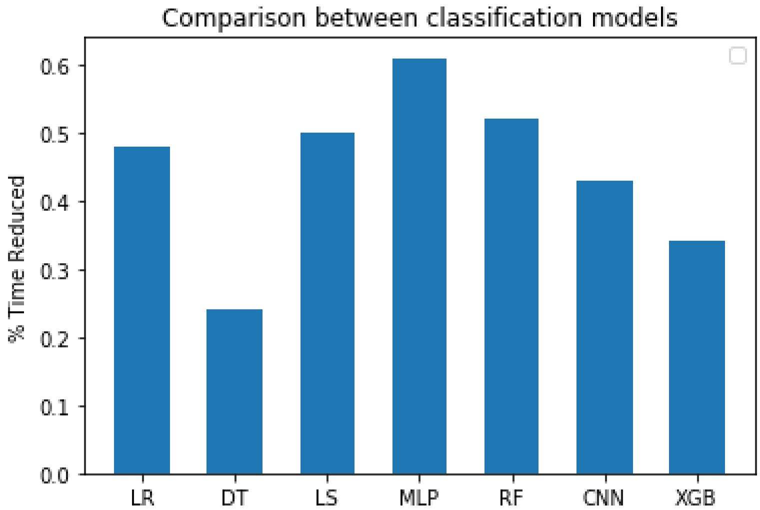


Figure 42. Efficiency gains for the smart home's activity classification

The results from the experiment indicate that approach has been unsuccessful in achieving an acceptable performance within a reasonable range of 5% from the baseline. The results confirm the suspicions that the novel lightweight ML method is less effective when dealing with non-linear data problems and requires an alternative solution. The data captured from the LPDs are non-linear, and most events, being captured, appear to be random. In other words, the events captured by the LPDs, such as using the toilet and relaxing, are inconsistent repetitive actions and are unlike activities such as walking, running and or swimming. The data set is, therefore considered non-linear, which required an alternative solution.

Further exploration of the data was required to establish if other useful information could be inferred to find a more suitable course of action. By encoding the activity target values, it was possible to examine the frequency probability distributions of each activity. The results confirmed the irregularities and the outputs were not normally distributed, and even those that were presumed regular did not output a bell-like curve. ***Figure 43*** shows the frequency distribution and QQ plots, where the mean $\mu = 4.87$ and the standard deviation $\sigma = 1.22$ for the sleeping activity and $\mu = 2.35$ $\sigma = 2.06$ for grooming. These values indicate a high skewness and suggest there are many outliers in the data. The QQ plot confirmed that a non-parametric approach was required.

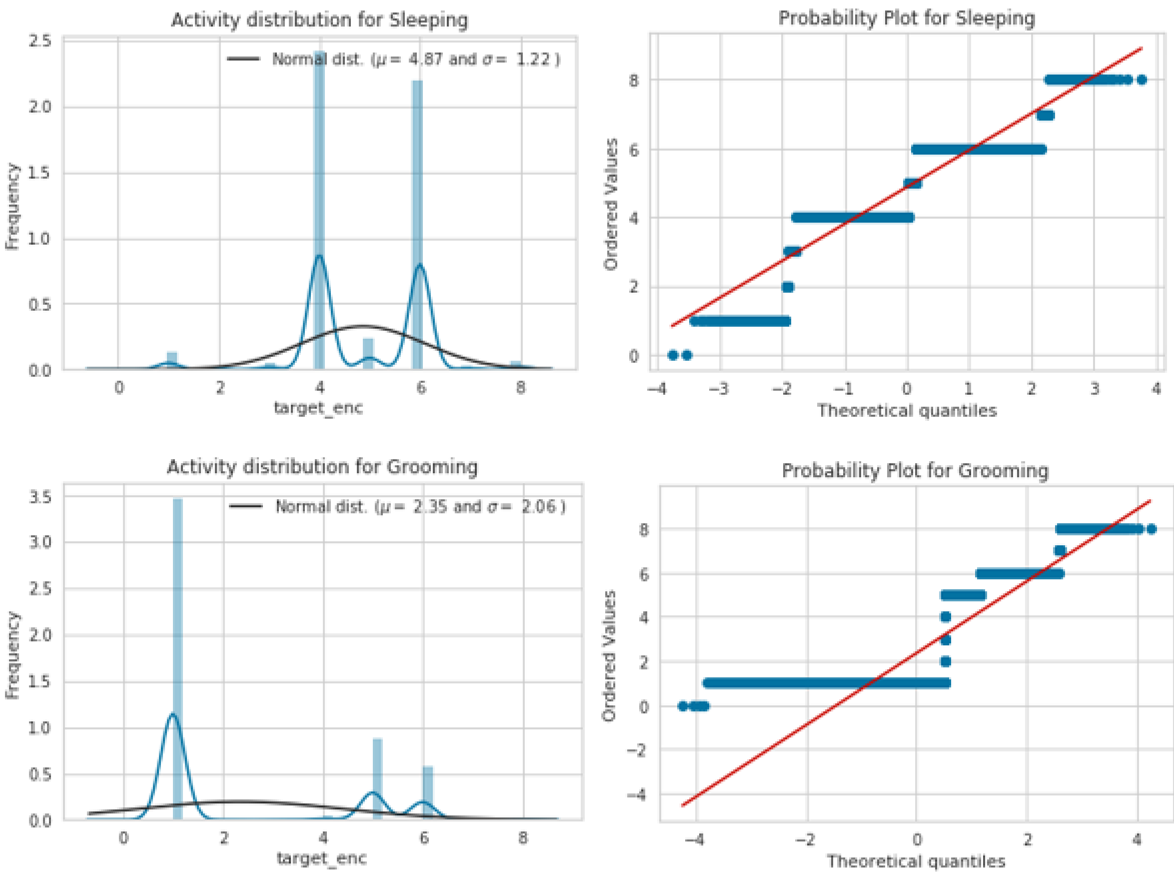


Figure 43. Frequency distribution and the probability quantile-quantile (Q-Q) plots for two activities

The QQ plots confirmed that a non-parametric approach was required, as the methods make fewer assumptions; it's necessary to adopt models that the structure of the relationship between variables are treated non-parametrically using statistical inference. The following models were chosen as per

M Eastwood et al. (2019). A Decision Tree: A single decision tree created by employing the split criterion and cost-complexity (M Eastwood et al., 2019). A Conditional Random Field (CRF) Template created using a set of simple template-based features. A CRF-Forest: created using features extracted from a Random Forest classifier. A CRF-Both created using features extracted from a Random Forest augmented with template-based features. A Multi-class SVM consisting of an ensemble of 1 vs 1 SVM classifiers. Finally, a Random Forest ensemble of 200 fully grown decision trees. All the models were trained using a random 70:30 split ration, 70% for training and 30% for testing.

The results of the multi-model Random Forest ensemble on the CASAS dataset for each activity is shown in ***Table 14***. To compare the results with the work carried out by Eastwood et al., 2019 is shown in ***Table 15***. The results show that using the diversified relevant complex features extracted using a random forest model produces the best performing CRF model. Four classifiers, Decision Tree, CRF-Forest, SVM and Random Forest, were evaluated as per Eastwood et al., 2019 experiments. The Decision Tree and SVM models showed marginal improvements, however, took significantly longer to train. The Random Forest model gained at least 1.4% 5 It appeared that when there were fewer outliers present in the CRF model, by removing the irregular activities from the model, it can better estimate the feature weights. Therefore, the additional features were unnecessarily complicating the optimisation process, as mentioned by Eastwood et al., 2019.

Table 14. The CFR-Forest average Zero-One Loss and 5-fold CV scores for individual activities

Activity	**Zero One Loss**	**5-Fold CV Score**
Sleeping	*0.076*	*85% (+/- 0.02)*
Grooming	*0.063*	*93.1% (+/- 0.01)*
Toilet	*0.198*	*61.2% (+/- 1.60)*
Eating	*0.068*	*85.01% (+/- 1.50)*
Washing	*0.080*	*82.02% (+/- 0.02)*
Relaxing	*0.137*	*63.06% (+/- 0.82)*
Working	*0.084*	*80.01% (+/- 0.35)*
Reading	*0.126*	*60.2% (+/- 0.46)*

Table 15. The average scores on Activity Recognition

Method	5-Fold CV Score
Decision Tree	*65.8% (+/- 1.98)*
CRF-Forest	*76.2% (+/- 1.61)*
SVM	*66.8% (+/- 2.72)*
Random Forest	*72.3% (+/- 1.80)*

7.2.3 Summary

During this exploratory experiment, it's concluded that the novel lightweight ML approach is ineffective when dealing with non-linear PA activity-based problems and significantly underperformed for all the ML models in the investigation. Efficiency gains have been achieved across all the ML models; however, it has rendered them unfeasible with the low classification performance. Instead, an alternative feature selection method has recommended for dealing with non-linear activities in a machine learning application that performs activity recognition using data gathered in a smart home setting. A non-parametric approach showed that using the diversified relevant complex features extracted using a random forest model and removing the model's irregular activities produces the best performing CRF model. The predictive performance gains using the features generated using the proposed method from the discrete activity groups was evident in the CRF-Forest model, which gained a 3% increase.

Chapter 8: Conclusions

8.1 Summary

A sedentary lifestyle is a genuine health concern. Researchers have produced evidence of regular Physical Activity (PA) helping to reduce the risk of a variety of health problems significantly, such as substantially reducing the risks associated with cancer (Murray et al., 2020), the treatment of asthma (Endre & DuBuske, 2018) and supporting injury recovery (Ekegren et al., 2020). Based on the compelling evidence in support of promoting regular PA, there is a need for monitoring and measuring suitable levels of PA activity is being performed. As over excessive PA can lead to unphysiological loads on the joints (Greca et al., 2019). Wearable devices have created a particular use case for researchers to monitor PA. Repetitive weight-bearing PA's such as walking, cycling and jumping activities are recommended for children to promote bone health (Greca et al., 2019; Landry & Driscoll, 2012) which can be monitored and measured using wearable devices. This research can benefit both practitioners and those who are health conscious for monitoring and maintaining a healthy, active lifestyle.

The thesis successfully achieved its objectives by investigating the use of various types of sensors in three different settings and evaluated their effectiveness in performing Activity Recognition (AR) using suitable machine learning models. The thesis presented empirical comparisons of the novel method and demonstrated its significance in three types of machine learning problems - binary classification, multi-class classification and regression problems. The initial findings of the method's effectiveness in selecting suitable features using open source automated supervised machine learning pipelines using three types of machine learning problems. The results showed that the technique was able to score within 5% of the baseline methods while also improving the ML model's efficiency by up to 70%, which is ideal for running ML models on low powered devices. The experiments also revealed that the performance gains were only prevalent on those datasets with large sample sizes and where there were monotonic relationships present between features of data sets linear in nature.

The first of three experimental case studies demonstrated the effective use of the proposed feature selection method in several supervised machine learning technique,s which showed that the energy expenditure could be predicted using the accelerometer readings from a wearable device. The study results were consistent with prior work undertaken in adults by Montoye, Westgate, et al. (2018), which reported that machine learning models predicted physical activities with an accuracy of 71-92% from wrist-worn GENEActiv accelerometers.

The second case study demonstrated the proposed feature selection method's effectiveness in selecting the suitable features for supervised machine learning models in swim activity recognition on the lower-powered edge device. The final case study provides key findings from the experimental exploration of non-linear based activity analysis. A spontaneous physical activity that does not produce a consistent signal is omnidirectional, such as playing football and playing catch is considered a non-linear activity. The research has confirmed the suspicion that the approach is not suitable for non-linear activities as expected.

This thesis has addressed its challenges and motivations by investigating and developing a lightweight automatic feature selection approach to deploy on low-powered devices. The approach focused on producing near real-time results to analyse time series-based data produced by wearable sensors during continuous linear physical activities. The thesis demonstrated the use of the lightweight approach running on a device with computationally limited resources while delivering accurate results within 5% of the baseline model.

The thesis has made the following contributions to the body of knowledge:

1. The research helped develop a novel automatic feature selection approach towards deploying lightweight ML applications on low powered devices for continuous linear repetitive physical activity.

2. The novel approach has been applied to a case study with linear based activities where supervised machine learning is used to accurately approximate children’s energy expenditure

or Metabolic Equivalents (MET). Body movement data is collected using wearable devices worn by children, emphasising physical activities that represent locomotor and object control movements commonly undertaken by children. The thesis discusses and compares the effectiveness of this method. The findings showed that the technique was able to score within 5% of the baseline methods while also improving the ML model's efficiency by up to 70%, which is ideal for running ML models on low powered devices.

3. The novel approach has also been applied in a case study with linear based activities where supervised machine learning is used in swim activity detection. The supervised machine learning methods used in this case study were Linear Regression, LSTM, kNN, Random Forests, Support Vector Machines, Boosted Trees and Neural Networks, the novel feature selection method successfully identified suitable features to train them. The thesis discusses and compares the effectiveness of this method. Machine learning models such as Random Forest, CNN, and Boosted Trees all performed above 90% accuracy in predicting energy expenditure. The computational efficiency gains were above 50% across all the models evaluated and up to 70% using Boosted Trees.

4. The novel lightweight approach has been applied to the CASAS smart homes case study with non-linear based activities. As anticipated, the results showed that the novel light approach was ineffective at appropriately classifying the activities to an acceptable level of performance. Instead, an alternative approach was recommended using a non-parametric method based on the initial experimental results.

8.2 Future Works

The focus of this thesis was around to improve the effectiveness of feature selection optimisation. And in doing so, a novel unified method that utilises ensemble machine learning techniques in sequential modelling for feature selection optimisation was developed. The method exploited the Spatio-temporal relations within data and was found to effective in activity-based recognition. However, the case study showed only marginal gains in accuracy performance, between 1% and 3%, for activity recognition. Future experimental studies may be conducted to evaluate semi-supervised classification models that may draw new inferences from specific temporal behaviours of individuals in the care home as opposed to the current single generalised model that attempts to fit all.

The methods predictive capabilities showed that the performance was simply on par with existing research and did not show any significant performance gains. This meant that even though that method was suitable for predicting energy expenditure, it was unable to confirm the best physical location for the wearable sensor. The future experimental analysis may focus on a better understanding of movement patterns of various activities in different environments to further improve the method's performance.

The classification of stroke types worked well for most of the machine learning models explored in this thesis. However, they all underperformed on turn detection timings. The turn detection algorithm was based on extracting the median derivative value found during the turn signal. Once the occurrence of turn was established, the problem of finding its exact time remained difficult. The algorithm then searched for the maximum probability that a turn signal was discovered and disregarded all others. However, failed to perform well on subsequent turns in the same session consistently and as a result, reduced the lap timing accuracies. It may have been possible to combine inputs from many turn signals to establish a normalised pattern.

The following potential approaches may be explored in future experiments:

1. Use the summing (integration) instead of the maximum.
2. Use another machine learning algorithm, i.e. gradient boosting to combine inputs.
3. Run another machine learning algorithm on the raw signal near the suspected turn to find better approximations of the exact turn start and end times.

A lower power Application-Specific Integrated Circuit (ASIC) called Edge TPU, developed by Google, allows high-performance ML interfacing which can run TensorFlow Lite locally on the device (Sakr et al., 2020; Sengupta et al., 2020; Subramanian et al., 2017). EdgeML developed by Microsoft which offers a suite of ML algorithms designed to work in severely resource-constrained scenarios such as LPDs. ARM has also published an open-source library called Cortex Microcontroller Software Interface Standard Neural Network (CMSIS-NN), for Cortex-M processors, which maximizes neural network performance. Finally, X-Cube-AI, has been developed for implementing deep learning models to be deployed on STM 32-bit microcontrollers. The potential for which should be explored to further enhance the performance and or efficiencies in running ML on the low power devices.

References:

Abidine, B. M., Fergani, L., Fergani, B., & Oussalah, M. (2018). The joint use of sequence features combination and modified weighted SVM for improving daily activity recognition. *Pattern Analysis and Applications*. https://doi.org/10.1007/s10044-016-0570-y

Acampora, G., Cook, D. J., Rashidi, P., & Vasilakos, A. V. (2013). A survey on ambient intelligence in healthcare. *Proceedings of the IEEE*. https://doi.org/10.1109/JPROC.2013.2262913

Agarwal, P., & Alam, M. (2020). A Lightweight Deep Learning Model for Human Activity Recognition on Edge Devices. *Procedia Computer Science*, *167*(2019), 2364–2373. https://doi.org/10.1016/j.procs.2020.03.289

Akinwande, M. O., Dikko, H. G., & Samson, A. (2015). Variance Inflation Factor: As a Condition for the Inclusion of Suppressor Variable(s) in Regression Analysis. *Open Journal of Statistics*, *05*(07), 754–767. https://doi.org/10.4236/ojs.2015.57075

Alinia, P., Saeedi, R., Mortazavi, B., Rokni, A., & Ghasemzadeh, H. (2015). Impact of sensor misplacement on estimating metabolic equivalent of task with wearables. *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks, BSN 2015*, 1–6. https://doi.org/10.1109/BSN.2015.7299385

Barten, A. P. (1987). The coefficient of determination for regression without a constant term. In R. Heijmans & H. Neudecker (Eds.), *The Practice of Econometrics: Studies on Demand, Forecasting, Money and Income* (pp. 181–189). https://doi.org/10.1007/978-94-009-3591-4_12

Bikmukhamedov, R. F., & Nadeev, A. F. (2019). Lightweight machine learning classifiers of IoT traffic flows. *2019 Systems of Signal Synchronization, Generating and Processing in Telecommunications, SYNCHROINFO 2019*, (December), 1–5. https://doi.org/10.1109/SYNCHROINFO.2019.8814156

Bischl, B., Kerschke, P., Kotthoff, L., Lindauer, M., Malitsky, Y., Fréchette, A., … Vanschoren, J. (2016). ASlib: A benchmark library for algorithm selection. *Artificial Intelligence*, *237*, 41–58. https://doi.org/10.1016/j.artint.2016.04.003

Blumchent, G. (1990). *Metabolic Equivalents (METS) in Exercise Testing , Exercise*

Prescription , and Evaluation of Functional Capacity. *565*, 555–565.

Blythe, A., Crouter, S. E., & LaMunion, S. R. (2017). *Validity of Consumer-Based Physical Activity Monitors for Estimating Energy Expenditure in Youth*. 1–30.

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2015). Recent advances and emerging challenges of feature selection in the context of big data. *Knowledge-Based Systems*, *86*, 33–45. https://doi.org/10.1016/j.knosys.2015.05.014

Bolón-Canedo, Verónica, Sechidis, K., Sánchez-Maroño, N., Alonso-Betanzos, A., & Brown, G. (2019). Insights into distributed feature ranking. *Information Sciences*, *496*, 378–398. https://doi.org/10.1016/j.ins.2018.09.045

Brodie, M. A., Pliner, E. M., Ho, A., Li, K., Chen, Z., Gandevia, S. C., & Lord, S. R. (2018). Big data vs accurate data in health research: Large-scale physical activity monitoring, smartphones, wearable devices and risk of unconscious bias. *Medical Hypotheses*, *119*(July), 32–36. https://doi.org/10.1016/j.mehy.2018.07.015

Cai, H., Xu, B., Jiang, L., & Vasilakos, A. V. (2017). IoT-Based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges. *IEEE Internet of Things Journal*, *4*(1), 75–87. https://doi.org/10.1109/JIOT.2016.2619369

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70–79. https://doi.org/10.1016/j.neucom.2017.11.077

Calvaresi, D., Cesarini, D., Sernani, P., Marinoni, M., Dragoni, A. F., & Sturm, A. (2017). Exploring the ambient assisted living domain: a systematic review. *Journal of Ambient Intelligence and Humanized Computing*, *8*(2), 239–257. https://doi.org/10.1007/s12652-016-0374-3

CASAS. (2015). Casas datasets. Retrieved March 2, 2019, from No. 24 website: http://casas.wsu.edu/datasets/

Chen, J., & Ran, X. (2019). Deep Learning With Edge Computing: A Review. *Proceedings of the IEEE*, *107*(8). https://doi.org/10.1109/JPROC.2019.2921977

Chen, K., Zhang, D., Yao, L., Guo, B., Yu, Z., & Liu, Y. (2020). *Deep Learning for Sensor-based Human Activity Recognition: Overview, Challenges and Opportunities*. *37*(4). Retrieved from http://arxiv.org/abs/2001.07416

Chen, W., Wang, D., & Li, K. (2019). Multi-User Multi-Task Computation Offloading in Green Mobile Edge Cloud Computing. *IEEE Transactions on Services Computing*, *12*(5), 726–738. https://doi.org/10.1109/TSC.2018.2826544

Chowdhury, E. A., Western, M. J., Nightingale, T. E., Peacock, O. J., & Thompson, D. (2017). Assessment of laboratory and daily energy expenditure estimates from consumer multisensor physical activity monitors. *PLoS ONE*, *12*(2), 1–15. https://doi.org/10.1371/journal.pone.0171720

Christofaro, D. G. D., Andersen, L. B., Andrade, S. M. de, Barros, M. V. G. de, Saraiva, B. T. C., Fernandes, R. A., & Ritti-Dias, R. M. (2018). Adolescents' physical activity is associated with previous and current physical activity practice by their parents. *Jornal de Pediatria*, *94*(1). https://doi.org/10.1016/j.jped.2017.01.007

Cook, D. J., Crandall, A. S., Thomas, B. L., & Krishnan, N. C. (2013). CASAS: A smart home in a box. *Computer*. https://doi.org/10.1109/MC.2012.328

Crouter, S. E., Oody, J. F., & Bassett, D. R. (2018). Estimating physical activity in youth using an ankle accelerometer. *Journal of Sports Sciences*. https://doi.org/10.1080/02640414.2018.1449091

Dahmen, J., Thomas, B. L., Cook, D. J., & Wang, X. (2017). Activity learning as a foundation for security monitoring in smart homes. *Sensors (Switzerland)*, *17*(4), 1–17. https://doi.org/10.3390/s17040737

Davis, J. J., & Foo, E. (2016). Automated feature engineering for HTTP tunnel detection. *Computers and Security*, *59*, 166–185. https://doi.org/10.1016/j.cose.2016.01.006

De Donno, M., Tange, K., & Dragoni, N. (2019). Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog. *IEEE Access*, *7*, 150936–150948. https://doi.org/10.1109/ACCESS.2019.2947652

Dennison, L., Morrison, L., Conway, G., & Yardley, L. (2013). Opportunities and Challenges for Smartphone Applications in Supporting Health Behavior Change: Qualitative Study. *J Med Internet Res*, *15*(4), e86. https://doi.org/10.2196/jmir.2583

Devakumari, D., & Thangavel, K. (2010). Unsupervised adaptive floating search feature selection based on Contribution Entropy. *2010 International Conference on Communication and Computational Intelligence (INCOCCI)*, 623–627.

Dheeru, D., & Karra Taniskidou, E. (2017). *{UCI} Machine Learning Repository*. Retrieved from http://archive.ics.uci.edu/ml

Dijkhuis, T. B., Blaauw, F. J., van Ittersum, M. W., Velthuijsen, H., & Aiello, M. (2018). Personalized physical activity coaching: A machine learning approach. *Sensors (Switzerland)*, *18*(2), 1–15. https://doi.org/10.3390/s18020623

Dipanjan, S., Raghav, B., & Sharma, T. (2018). *Practical Machine Learning with Python*. Retrieved from https://doi.org/10.1007/978-1-4842-3207-1

Dormann, C. F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., … Lautenbach, S. (2013). Collinearity: A review of methods to deal with it and a simulation study evaluating their performance. *Ecography*, *36*(1), 027–046. https://doi.org/10.1111/j.1600-0587.2012.07348.x

Duncan, M. J., Roscoe, C. M. P., Faghy, M., Tallis, J., & Eyre, E. L. J. (2019). Estimating Physical Activity in Children Aged 8–11 Years Using Accelerometry: Contributions From Fundamental Movement Skills and Different Accelerometer Placements. *Frontiers in Physiology*, *10*. https://doi.org/10.3389/fphys.2019.00242

Duncan, M. J., Wilson, S., Tallis, J., & Eyre, E. (2016). Validation of the Phillips et al. GENEActiv accelerometer wrist cut-points in children aged 5–8 years old. *European Journal of Pediatrics*. https://doi.org/10.1007/s00431-016-2795-6

Eastwood, M, Konios, A., Tan, B., Jing, Y., & Hamid, A. (2019). Conditional Random Field Feature Generation of Smart Home Sensor Data using Random Forests. *2019 International Microwave Biomedical Conference (IMBioC 2019)*.

Eastwood, Mark, Konios, A., Tan, B., Jing, Y., & Hamid, A. (2019). Conditional Random Field Feature Generation of Smart Home Sensor Data using Random Forests. *2019 International Microwave Biomedical Conference (IMBioC 2019)*, *(In-press)*, (In--press). IEEE.

Ekegren, C. L., Braaf, S., Ameratunga, S., Ponsford, J., Nunn, A., Cameron, P., … Gabbe, B. J. (2020). Adaptation, self-motivation and support services are key to physical activity participation three to five years after major trauma: a qualitative study. *Journal of Physiotherapy*. https://doi.org/10.1016/j.jphys.2020.06.008

Elkan, C. (2010). Predictive analytics and data mining. In *Npl*. Retrieved from http://www.mendeley.com/research/data-mining-and-predictive-analysis/

Ellis, D. A., & Piwek, L. (2018). Failing to encourage physical activity with wearable technology: what next? *Journal of the Royal Society of Medicine*, *111*(9), 310–313. https://doi.org/10.1177/0141076818788856

Endre, L., & DuBuske, L. (2018). IMPACT OF SWIMMING THERAPY AND PHYSICAL ACTIVITY ON ASTHMA. *Annals of Allergy, Asthma & Immunology*, *121*(5, Supplement), S42. https://doi.org/https://doi.org/10.1016/j.anai.2018.09.135

Escamilla-Ambrosio, P. J., Rodríguez-Mota, A., Aguirre-Anaya, E., Acosta-Bermejo, R., & Salinas-Rosales, M. (2018). Distributing computing in the internet of things: Cloud, fog and edge computing overview. *Studies in Computational Intelligence*, *731*, 87–115. https://doi.org/10.1007/978-3-319-64063-1_4

Esliger, D. W., Rowlands, A. V., Hurst, T. L., Catt, M., Murray, P., & Eston, R. G. (2011). Validation of the GENEA accelerometer. *Medicine and Science in Sports and Exercise*.

https://doi.org/10.1249/MSS.0b013e31820513be

Fani, H., Mirlohi, A., Hosseini, H., & Herperst, R. (2018). Swim stroke analytic: Front crawl pulling pose classification. *Proceedings - International Conference on Image Processing, ICIP*, 4068–4072. https://doi.org/10.1109/ICIP.2018.8451756

Farrahi, V., Niemelä, M., Kangas, M., Korpelainen, R., & Jämsä, T. (2019). Calibration and validation of accelerometer-based activity monitors: A systematic review of machine-learning approaches. *Gait and Posture*, *68*(November 2018), 285–299. https://doi.org/10.1016/j.gaitpost.2018.12.003

Fergus, P., Hussain, A. J., Hearty, J., Fairclough, S., Boddy, L., Mackintosh, K., … Lunn, J. (2017). A machine learning approach to measure and monitor physical activity in children. *Neurocomputing*, *228*(November 2016), 220–230. https://doi.org/10.1016/j.neucom.2016.10.040

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015a). Efficient and Robust Automated Machine Learning. *Advances in Neural Information Processing Systems 28*, 2944–2952. Retrieved from http://papers.nips.cc/paper/5872-efficient-and-robust-automated-machine-learning.pdf

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J. T., Blum, M., & Hutter, F. (2015b). Methods for Improving Bayesian Optimization for AutoML. *ICML 2015 AutoML Workshop*, (Section 3).

Fine, S., Singer, Y., & Tishby, N. (1998). The hierarchical hidden Markov model: Analysis and applications. *Machine Learning*, *32*(1), 41–62. https://doi.org/10.1023/A:1007469218079

Finlay, S. (2014). Introduction. In: Predictive Analytics, Data Mining and Big Data. Business in the Digital Economy. *Palgrave Macmillan, London*, 1–2.

Gil-Martín, M., San-Segundo, R., Fernández-Martínez, F., & Ferreiros-López, J. (2020). Improving physical activity recognition using a new deep learning architecture and post-processing techniques. *Engineering Applications of Artificial Intelligence*, *92*(March), 103679. https://doi.org/10.1016/j.engappai.2020.103679

Gjoreski, H., Kaluža, B., Gams, M., Milić, R., & Luštrek, M. (2013). *Ensembles of multiple sensors for human energy expenditure estimation*. 359. https://doi.org/10.1145/2493432.2493517

Greca, J. P. de A., Ryan, J., Baltzopoulos, V., & Korff, T. (2019). Biomechanical evaluation of walking and cycling in children. *Journal of Biomechanics*, *87*, 13–18. https://doi.org/10.1016/j.jbiomech.2019.01.051

Greco, L., Percannella, G., Ritrovato, P., Tortorella, F., & Vento, M. (2020). Trends in IoT based solutions for health care: Moving AI to the edge. *Pattern Recognition Letters*, *135*, 346–353. https://doi.org/10.1016/j.patrec.2020.05.016

Guo, Y., Jia, X., & Paull, D. (2018). Effective Sequential Classifier Training for SVM-Based Multitemporal Remote Sensing Image Classification. *IEEE Transactions on Image Processing*, *27*(6), 3036–3048. https://doi.org/10.1109/TIP.2018.2808767

Guyon, I., Bennett, K., Cawley, G., Escalante, H. J., Escalera, S., Ho, T. K., … Viegas, E. (2015). Design of the 2015 ChaLearn AutoML challenge. *Proceedings of the International Joint Conference on Neural Networks*, *2015-Septe*. https://doi.org/10.1109/IJCNN.2015.7280767

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research (JMLR)*, *3*(3), 1157–1182. https://doi.org/10.1016/j.aca.2011.07.027

H.K. Montoye, A., M. Pivarnik, J., M. Mudd, L., Biswas, S., & A. Pfeiffer, K. (2016). Validation and Comparison of Accelerometers Worn on the Hip, Thigh, and Wrists for Measuring Physical Activity and Sedentary Behavior. *AIMS Public Health*, *3*(2), 298–312. https://doi.org/10.3934/publichealth.2016.2.298

Hakozaki, K., Kato, N., Tanabiki, M., Furuyama, J., Sato, Y., & Aoki, Y. (2018). Swimmer's Stroke Estimation Using CNN and MultiLSTM. *Journal of Signal Processing*, *22*(4), 219–222. https://doi.org/10.2299/jsp.22.219

Harrell, J. S., Mcmurray, R. G., Baggett, C. D., Pennell, M. L., Pearce, P. F., & Bangdiwala, S. I. (2005). Energy costs of physical activities in children and adolescents. *Medicine and Science in Sports and Exercise*, *37*(2), 329–336. https://doi.org/10.1249/01.MSS.0000153115.33762.3F

Hassan, M. M., Uddin, M. Z., Mohamed, A., & Almogren, A. (2018). A robust human activity recognition system using smartphone sensors and deep learning. *Future Generation Computer Systems*, *81*, 307–313. https://doi.org/10.1016/j.future.2017.11.029

Helal, S., & Bull, C. N. (2019). From Smart Homes to Smart-Ready Homes and Communities. *Dementia and Geriatric Cognitive Disorders*. https://doi.org/10.1159/000497803

Hindawi, M., Elghazel, H., & Benabdeslem, K. (2013). *Efficient semi-supervised feature selection by an ensemble approach*.

Höglund, H. (2017). Tax payment default prediction using genetic algorithm-based variable selection. *Expert Systems with Applications*, *88*(1), 368–375. https://doi.org/10.1016/j.eswa.2017.07.027

Holfelder, B., & Schott, N. (2014). Relationship of fundamental movement skills and physical activity in children and adolescents: A systematic review. *Psychology of Sport and Exercise*. https://doi.org/10.1016/j.psychsport.2014.03.005

Hu, Y., & Huang, S. (2017). *Challenges of Automated Machine Learning on Causal Impact Analytics for Policy Evaluation*.

Huddleston, S. H., & Brown, G. G. (2019). *Machine Learning 7 . 2 Supervised , Unsupervised , and Reinforcement Learning*. 231–274.

Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6683 LNCS,* 507–523. https://doi.org/10.1007/978-3-642-25566-3_40

Inoubli, W., Aridhi, S., Mezni, H., Maddouri, M., & Mephu Nguifo, E. (2018). An experimental survey on big data frameworks. *Future Generation Computer Systems*, *86*, 546–564. https://doi.org/10.1016/j.future.2018.04.032

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., … Kalenichenko, D. (2018). Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2704–2713. https://doi.org/10.1109/CVPR.2018.00286

Jing, Y., Eastwood, M., Tan, B., Konios, A., Hamid, A., & Collinson, M. (2017). An Intelligent Well-being Monitoring System for Residents in Extra Care Homes. *Proceedings of the 1st International Conference on Internet of Things and Machine Learning*, 8:1--8:6. https://doi.org/10.1145/3109761.3109769

Kanter, J. M., & Veeramachaneni, K. (2015). *Deep Feature Synthesis:Towards Automating Data Science Endeavors*. 1–10. https://doi.org/10.1109/DSAA.2015.7344858

Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., & Chouvarda, I. (2017). Machine Learning and Data Mining Methods in Diabetes Research. *Computational and Structural Biotechnology Journal*. https://doi.org/10.1016/j.csbj.2016.12.005

Klein, A., Falkner, S., Mansur, N., & Hutter, F. (2017). RoBO: A Flexible and Robust Bayesian Optimization Framework in Python. *NIPS Bayesian Optimization Workshop*, (Nips). Retrieved from https://bayesopt.github.io/papers/2017/22.pdf

Kobayashi, M., OMAE, Y., SAKAI, K., AKIDUKI, T., SHIONOYA, A., & TAKAHASHI, H. (2019). Estimation Method of Turn Section During Swimming by Using Ensemble Learning and Single Inertial Sensor単一慣性センサとアンサンブル学習を活用した

競泳中のターン区間推定. *Journal of Japan Society for Fuzzy Theory and Intelligent Informatics*, *31*(1), 597–602. https://doi.org/10.3156/jsoft.31.1_597

Konios, A., Jing, Y., Eastwood, M., & Tan, B. (2018). *Unifying and Analysing Activities of Daily Living in Extra Care Homes*. Retrieved from http://cyber-science.org/2018/

Kos, A., & Umek, A. (2019). Wearable Sensor Devices for Prevention and Rehabilitation in Healthcare: Swimming Exercise With Real-Time Therapist Feedback. *IEEE Internet of Things Journal*, *6*(2), 1331–1341. https://doi.org/10.1109/JIOT.2018.2850664

Kotthoff, L., Gent, I. P., & Miguel, I. (2011). An Evaluation of Machine Learning in Algorithm Selection for Search Problems. *The Fourth International Symposium on Combinatorial Search*, *25*(3), 257–270. https://doi.org/10.3233/AIC-2012-0533

Kotu, V., & Deshpande, B. (2015). Data Mining Process. *Predictive Analytics and Data Mining*, (1), 17–36. https://doi.org/10.1016/B978-0-12-801460-8.00002-1

Kumar, V. (2014). Feature Selection: A literature Review. *The Smart Computing Review*, *4*(3). https://doi.org/10.6029/smartcr.2014.03.007

Lafferty, J. D., McCallum, A., & Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the Eighteenth International Conference on Machine Learning*.

Landry, B. W., & Driscoll, S. W. (2012). Physical Activity in Children and Adolescents. *PM&R*, *4*(11), 826–832. https://doi.org/https://doi.org/10.1016/j.pmrj.2012.09.585

Lebherz, D. S., Lorig, F., & Timm, I. J. (2019). Agent-based modeling and simulation of individual elderly care decision-making. *Proceedings - Winter Simulation Conference*, *2018-Decem*, 1025–1036. https://doi.org/10.1109/WSC.2018.8632293

Lechevalier, D., Narayanan, A., & Rachuri, S. (2015). Towards a domain-specific framework for predictive analytics in manufacturing. *Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014*, 987–995. https://doi.org/10.1109/BigData.2014.7004332

Lee, S. J., Yoo, P. D., Taufiq Asyhari, A., Jhi, Y., Chermak, L., Yeun, C. Y., & Taha, K. (2020). IMPACT: Impersonation Attack Detection via Edge Computing Using Deep Autoencoder and Feature Abstraction. *IEEE Access*, *8*. https://doi.org/10.1109/ACCESS.2020.2985089

Leppänen, T., Savaglio, C., & Fortino, G. (2020). Service modeling for opportunistic edge computing systems with feature engineering. *Computer Communications*, *157*(April), 308–319. https://doi.org/10.1016/j.comcom.2020.04.011

Li, J., & Liu, H. (2017). Challenges of Feature Selection for Big Data Analytics. *IEEE*

Intelligent Systems, *32*(2), 9–15. https://doi.org/10.1109/MIS.2017.38

Li, M., Gu, W., Chen, W., He, Y., Wu, Y., & Zhang, Y. (2018). Smart Home: Architecture, Technologies and Systems. *Procedia Computer Science*, *131*, 393–400. https://doi.org/10.1016/J.PROCS.2018.04.219

Liciotti, D., Bernardini, M., Romeo, L., & Frontoni, E. (2019). A sequential deep learning application for recognising human activities in smart homes. *Neurocomputing*, (xxxx). https://doi.org/10.1016/j.neucom.2018.10.104

Lindauer, M., Bergdoll, R. D., & Hutter, F. (2016). An empirical study of per-instance algorithm scheduling. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10079 LNCS*, 253–259. https://doi.org/10.1007/978-3-319-50349-3_20

Lindauer, M., Hutter, F., Hoos, H. H., & Schaub, T. (2017). AutoFolio: An automatically configured algorithm selector. *IJCAI International Joint Conference on Artificial Intelligence*, 5025–5029. https://doi.org/10.1613/jair.4726

Lu, H., Gu, C., Luo, F., Ding, W., & Liu, X. (2020). Optimization of lightweight task offloading strategy for mobile edge computing based on deep reinforcement learning. *Future Generation Computer Systems*, *102*, 847–861. https://doi.org/10.1016/j.future.2019.07.019

Lu, W., & Tan, Y. P. (2002). Swimmer motion analysis with application to drowning detection. *Proceedings - IEEE International Symposium on Circuits and Systems*, *2*, 660–663. https://doi.org/10.1109/iscas.2002.1011439

Lyden, K., Keadle, S. K., Staudenmayer, J., Freedson, P., & Alhassan, S. (2013). Energy Cost of Common Activities in Children and Adolescents. *Journal of Physical Activity and Health*, *10*(1), 62–69. https://doi.org/10.1123/jpah.10.1.62

Manini, T., Seguin, R., Buchner, D. M., Liu, J., Allison, M., Wang, C. Y., … Lacroix, A. Z. (2014). Sedentary behavior and mortality in older women: The women's health initiative. *American Journal of Preventive Medicine*. https://doi.org/10.1016/j.amepre.2013.10.021

Marinho, D., Barbosa, T., Neiva, H., Silva, A., & Morais, J. (2020). Comparison of the Start, Turn and Finish Performance of Elite Swimmers in 100 m and 200 m Races. *Journal of Sports Science & Medicine*, 397–407.

Mehra, N., & Gupta, S. (2013). Survey on Multiclass Classification Methods. *International Journal of Computer Science and Information Technologies*.

Molina, L., Belanche, L., & Nebot, À. (2002). Feature selection algorithms: A survey and experimental evaluation. *Data Mining, 2002. ICDM …*.

https://doi.org/10.1109/ICDM.2002.1183917

Montoye, A. H. K., Moore, R. W., Bowles, H. R., Korycinski, R., & Pfeiffer, K. A. (2018). Reporting accelerometer methods in physical activity intervention studies: A systematic review and recommendations for authors. *British Journal of Sports Medicine*. https://doi.org/10.1136/bjsports-2015-095947

Montoye, A. H. K., Mudd, L. M., Biswas, S., & Pfeiffer, K. A. (2015). Energy expenditure prediction using raw accelerometer data in simulated free living. *Medicine and Science in Sports and Exercise*, *47*(8), 1735–1746. https://doi.org/10.1249/MSS.0000000000000597

Montoye, A. H. K., Westgate, B. S., Fonley, M. R., & Pfeiffer, K. A. (2018). Cross-validation and out-of-sample testing of physical activity intensity predictions with a wrist-worn accelerometer. *Journal of Applied Physiology*, *124*(5), 1284–1293. https://doi.org/10.1152/japplphysiol.00760.2017

Mooney, R., Corley, G., Godfrey, A., Quinlan, L. R., & ÓLaighin, G. (2015). Inertial sensor technology for elite swimming performance analysis: A systematic review. *Sensors (Switzerland)*, *16*(1). https://doi.org/10.3390/s16010018

Mooney, R., Quinlan, L. R., Corley, G., Godfrey, A., Osborough, C., & ÓLaighin, G. (2017). Evaluation of the Finis Swimsense® and the Garmin Swim™ activity monitors for swimming performance and stroke kinematics analysis. *PLoS ONE*, *12*(2), 0–17. https://doi.org/10.1371/journal.pone.0170902

Murray, J. M., Coleman, H. G., & Hunter, R. F. (2020). Physical activity and cancer risk: Findings from the UK Biobank, a large prospective cohort study. *Cancer Epidemiology*, *68*(June), 101780. https://doi.org/10.1016/j.canep.2020.101780

Narimatsu, H., & Kasai, H. (2015). Duration and Interval Hidden Markov Model for sequential data analysis. *Proceedings of the International Joint Conference on Neural Networks*, *2015-Septe*. https://doi.org/10.1109/IJCNN.2015.7280808

Ni, J., Zhang, K., Lin, X., & Shen, X. S. (2018). Securing Fog Computing for Internet of Things Applications: Challenges and Solutions. *IEEE Communications Surveys and Tutorials*, *20*(1), 601–628. https://doi.org/10.1109/COMST.2017.2762345

Nisbet, R., Miner, G., Yale, K., Nisbet, R., Miner, G., & Yale, K. (2018a). A Data Preparation Cookbook. *Handbook of Statistical Analysis and Data Mining Applications*, 727–740. https://doi.org/10.1016/B978-0-12-416632-5.00018-9

Nisbet, R., Miner, G., Yale, K., Nisbet, R., Miner, G., & Yale, K. (2018b). Chapter 3 - The Data Mining and Predictive Analytic Process. *Handbook of Statistical Analysis and Data Mining Applications*, 39–54. https://doi.org/10.1016/B978-0-12-416632-5.00003-7

Nithya, B., & Ilango, V. (2017). Predictive analytics in health care using machine learning tools and techniques. *Proceedings of the 2017 International Conference on Intelligent Computing and Control Systems, ICICCS 2017*, *2018-Janua*, 492–499. https://doi.org/10.1109/ICCONS.2017.8250771

Novakovic, J. (2009). Using Information Gain Attribute Evaluation to Classify Sonar Targets. *17 ThT Elecommunucation Forum*, 1351–1354.

Olson, R. S., Bartley, N., Urbanowicz, R. J., & Moore, J. H. (2016). *Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science*. 485–492. https://doi.org/10.1145/2908812.2908918

Omae, Y., Kobayashi, M., Sakai, K., Akiduki, T., Shionoya, A., & Takahashi, H. (2020). Detection of swimming stroke start timing by deep learning from an inertial sensor. *ICIC Express Letters, Part B: Applications*, *11*(3), 245–251. https://doi.org/10.24507/icicelb.11.03.245

Omae, Y., Kon, Y., Kobayashi, M., Sakai, K., Shionoya, A., Takahashi, H., … Miyaji, C. (2017). Swimming Style Classification Based on Ensemble Learning and Adaptive Feature Value by Using Inertial Measurement Unit. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, *21*, 616–631. https://doi.org/10.20965/jaciii.2017.p0616

Onggo, B. S. S. (2012). Simulation modeling in the social care sector: A literature review. *Proceedings - Winter Simulation Conference*, 1–12. https://doi.org/10.1109/WSC.2012.6465275

Papagiannaki, A., Zacharaki, E. I., Kalouris, G., Kalogiannis, S., Deltouzos, K., Ellul, J., & Megalooikonomou, V. (2019). Recognizing physical activity of older people from wearable sensors and inconsistent data. *Sensors (Switzerland)*, *19*(4). https://doi.org/10.3390/s19040880

Phillips, L. R. S., Parfitt, G., & Rowlands, A. V. (2013). Calibration of the GENEA accelerometer for assessment of physical activity intensity in children. *Journal of Science and Medicine in Sport*. https://doi.org/10.1016/j.jsams.2012.05.013

Preece, S. J., Goulermas, J. Y., Kenney, L. P. J., Howard, D., Meijer, K., & Crompton, R. (2009). Activity identification using body-mounted sensors - A review of classification techniques. *Physiological Measurement*, *30*(4). https://doi.org/10.1088/0967-3334/30/4/R01

Punithavathi, P., Geetha, S., Karuppiah, M., Islam, S. H., Hassan, M. M., & Choo, K. K. R. (2019). A lightweight machine learning-based authentication framework for smart IoT devices. *Information Sciences*, *484*, 255–268. https://doi.org/10.1016/j.ins.2019.01.073

Qi, M., Wang, T., Liu, F., Zhang, B., Wang, J., & Yi, Y. (2018). Unsupervised feature selection by regularized matrix factorization. *Neurocomputing*, *273*, 593–610. https://doi.org/10.1016/j.neucom.2017.08.047

Qian, M., & Zhai, C. (2013). Robust unsupervised feature selection. *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, 1621–1627. Retrieved from http://dl.acm.org/citation.cfm?id=2540361

Rahmani, A. M., Gia, T. N., Negash, B., Anzanpour, A., Azimi, I., Jiang, M., & Liljeberg, P. (2018). Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach. *Future Generation Computer Systems*, *78*, 641–658. https://doi.org/10.1016/j.future.2017.02.014

Rajput, A. S., Raman, B., & Imran, J. (2020). Privacy-preserving human action recognition as a remote cloud service using RGB-D sensors and deep CNN. *Expert Systems with Applications*, *152*, 113349. https://doi.org/10.1016/j.eswa.2020.113349

Roscoe, C. M. P., James, R. S., & Duncan, M. J. (2017). Calibration of GENEActiv accelerometer wrist cut-points for the assessment of physical activity intensity of preschool aged children. *European Journal of Pediatrics*. https://doi.org/10.1007/s00431-017-2948-2

Routen, A. C., Upton, D., Edwards, M. G., & Peters, D. M. (2012). Discrepancies in accelerometer-measured physical activity in children due to cut-point non-equivalence and placement site. *Journal of Sports Sciences*, *30*(12), 1303–1310. https://doi.org/10.1080/02640414.2012.709266

Rowlands, A. V., & Eston, R. G. (2007). The measurement and interpretation of children's physical activity. *Journal of Sports Science and Medicine*.

Rowlands, Alex V., Phillips, L. R. S., & Parfitt, G. (2013). Calibration of the GENEA accelerometer for assessment of physical activity intensity in children. *Journal of Science and Medicine in Sport*. https://doi.org/10.1016/j.jsams.2012.05.013

Ryan, J., & Gormley, J. (2013). An evaluation of energy expenditure estimation by three activity monitors. *European Journal of Sport Science*. https://doi.org/10.1080/17461391.2013.776639

Sacko, R. S., McIver, K., Brian, A., & Stodden, D. F. (2018). New insight for activity intensity relativity, metabolic expenditure during object projection skill performance. *Journal of Sports Sciences*. https://doi.org/10.1080/02640414.2018.1459152

Saez, Y., Baldominos, A., & Isasi, P. (2017). A comparison study of classifier algorithms for cross-person physical activity recognition. *Sensors (Switzerland)*, *17*(1), 1–26. https://doi.org/10.3390/s17010066

Sakr, F., Bellotti, F., Berta, R., & De Gloria, A. (2020). Machine learning on mainstream microcontrollers†. *Sensors (Switzerland)*, *20*(9). https://doi.org/10.3390/s20092638

Samie, F., Bauer, L., & Henkel, J. (2019). From cloud down to things: An overview of machine learning in internet of things. *IEEE Internet of Things Journal*, *6*(3), 4921–4934. https://doi.org/10.1109/JIOT.2019.2893866

Samie, F., Tsoutsouras, V., Bauer, L., Xydis, S., Soudris, D., & Henkel, J. (2019). Oops: Optimizing operation-mode selection for IoT edge devices. *ACM Transactions on Internet Technology*, *19*(2). https://doi.org/10.1145/3230642

Schwab, A., & Starbuck, W. H. (2013). Why baseline modelling is better than null-hypothesis testing: Examples from international business research. *Advances in International Management*, *26*, 171–195. https://doi.org/10.1108/S1571-5027(2013)0000026012

Sengupta, J., Kubendran, R., Neftci, E., & Andreou, A. (2020). High-Speed, Real-Time, Spike-Based Object Tracking and Path Prediction on Google Edge TPU. *Proceedings - 2020 IEEE International Conference on Artificial Intelligence Circuits and Systems, AICAS 2020*. https://doi.org/10.1109/AICAS48895.2020.9073867

Shafiq, M., Tian, Z., Sun, Y., Du, X., & Guizani, M. (2020). Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city. *Future Generation Computer Systems*, *107*, 433–442. https://doi.org/10.1016/j.future.2020.02.017

Sheikhpour, R., Sarram, M. A., Gharaghani, S., & Chahooki, M. A. Z. (2017). A Survey on semi-supervised feature selection methods. *Pattern Recognition*, *64*(February 2016), 141–158. https://doi.org/10.1016/j.patcog.2016.11.003

Sheiner, L. B., & Beal, S. L. (1981). Some suggestions for measuring predictive performance. *Journal of Pharmacokinetics and Biopharmaceutics*, *9*(4), 503–512. https://doi.org/10.1007/BF01060893

Siirtola, P., Laurinen, P., Roning, J., & Kinnunen, H. (2011). Efficient accelerometer-based swimming exercise tracking. *IEEE SSCI 2011: Symposium Series on Computational Intelligence - CIDM 2011: 2011 IEEE Symposium on Computational Intelligence and Data Mining*, 156–161. https://doi.org/10.1109/CIDM.2011.5949430

Solorio-Fernández, S., Carrasco-Ochoa, J. A., & Martínez-Trinidad, J. F. (2020). A review of unsupervised feature selection methods. *Artificial Intelligence Review*, *53*(2), 907–948. https://doi.org/10.1007/s10462-019-09682-y

Subramanian, S., Mustamo, P., Pettersson, D., Nyquist, R., Impiombato, D., Giarrusso, S., … Doha, T. (2017). Mobile Object Detection using TensorFlow Lite and Transfer Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *1004*(1), 7263–7271. https://doi.org/10.5339/qfarc.2018.ictpp417

Sutton, C. (2012). An Introduction to Conditional Random Fields. *Foundations and Trends® in Machine Learning*, *4*(4), 267–373. https://doi.org/10.1561/2200000013

Tang, Y., Teng, Q., Zhang, L., Min, F., & He, J. (2020, May 8). Efficient convolutional neural networks with smaller filters for human activity recognition using wearable sensors. *ArXiv*. arXiv.

Tsochantaridis, I., Hofmann, T., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large Margin Methods for Structured and Interdependent Output Variables. *Journal of Machine Learning Research (JMLR)*. https://doi.org/10.1007/s10994-008-5071-9

Uysal, A. K. (2016). An improved global feature selection scheme for text classification. *Expert Systems with Applications*, *43*, 82–92. https://doi.org/10.1016/j.eswa.2015.08.050

Van Kasteren, T. L. M., Englebienne, G., & Kröse, B. J. A. (2010). Activity recognition using semi-Markov models on real world smart home datasets. *Journal of Ambient Intelligence and Smart Environments*. https://doi.org/10.3233/AIS-2010-0070

Wang, Xiaoliang, Bai, L., Yang, Q., Wang, L., & Jiang, F. (2019). A dual privacy-preservation scheme for cloud-based eHealth systems. *Journal of Information Security and Applications*, *47*, 132–138. https://doi.org/10.1016/j.jisa.2019.04.010

Wang, Xiaying, Magno, M., Cavigelli, L., & Benini, L. (2020). FANN-on-MCU: An Open-Source Toolkit for Energy-Efficient Neural Network Inference at the Edge of the Internet of Things. *IEEE Internet of Things Journal*, *7*(5), 4403–4417. https://doi.org/10.1109/JIOT.2020.2976702

Waring, J., Lindvall, C., & Umeton, R. (2020). Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artificial Intelligence in Medicine*, *104*(October 2019), 101822. https://doi.org/10.1016/j.artmed.2020.101822

Wever, M., Mohr, F., & Hüllermeier, E. (2018). ML-Plan for Unlimited-Length Machine Learning Pipelines. *Proceedings of the International Workshop on Automatic Machine Learning*, (August).

Wonsuk, Y., Robert, M., Sejong, B., Karan, S., Qinghua, (Peter) He, & James, W. L. J. (2013). A Study of Effects of MultiCollinearity in the Multivariable Analysis. *International Journal of Applied Science and Technology*, *6*(8), 9–19. https://doi.org/10.1021/nn300902w.Release

Xu, L., Hutter, F., Hoos, H., & Leyton-brown, K. (2008). SATzilla: Portfolio-based Algorithm Selection for SAT. *J. Artif. Intell. Res.(JAIR)*, *32*, 565–606. https://doi.org/http://dx.doi.org/10.1613/jair.2490

Yao, S., Hu, S., Zhao, Y., Zhang, A., & Abdelzaher, T. (2017). DeepSense: A unified deep learning framework for time-series mobile sensing data processing. *26th International World Wide Web Conference, WWW 2017*, 351–360. https://doi.org/10.1145/3038912.3052577

Yavelberg, L., Zaharieva, D., Cinar, A., Riddell, M. C., & Jamnik, V. (2018). A Pilot Study Validating Select Research-Grade and Consumer-Based Wearables Throughout a Range of Dynamic Exercise Intensities in Persons With and Without Type 1 Diabetes: A Novel Approach. *Journal of Diabetes Science and Technology*, *12*(3), 569–576. https://doi.org/10.1177/1932296817750401

Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., … Jue, J. P. (2019). All one needs to know about fog computing and related edge computing paradigms: A complete survey. *Journal of Systems Architecture*, *98*(December 2018), 289–330. https://doi.org/10.1016/j.sysarc.2019.02.009

Z.M., H., & D.F., G. (2015). A review of feature selection and feature extraction methods applied on microarray data. *Advances in Bioinformatics*, *2015*(1). https://doi.org/10.1155/2015/198363

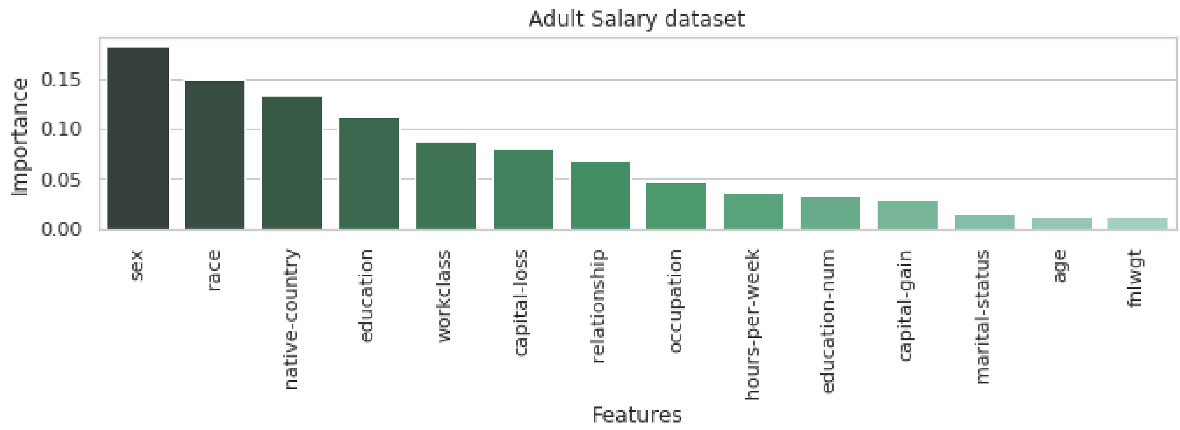
Zeng, Z., Zhang, H., Zhang, R., & Yin, C. (2015). A novel feature selection method considering feature interaction. *Pattern Recognition*, *48*(8), 2656–2666. https://doi.org/10.1016/j.patcog.2015.02.025

Zhang, B., Han, J., Huang, Z., Yang, J., & Zeng, X. (2019). A Real-Time and Hardware-Efficient Processor for Skeleton-Based Action Recognition with Lightweight Convolutional Neural Network. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *66*(12), 2052–2056. https://doi.org/10.1109/TCSII.2019.2899829

Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, *237*, 350–361. https://doi.org/10.1016/j.neucom.2017.01.026

Appendix 1:

Feature Importance of each binary classification dataset, baseline features are in Green and after applying the proposed improved feature selection method are in Red to Blue, respectively.



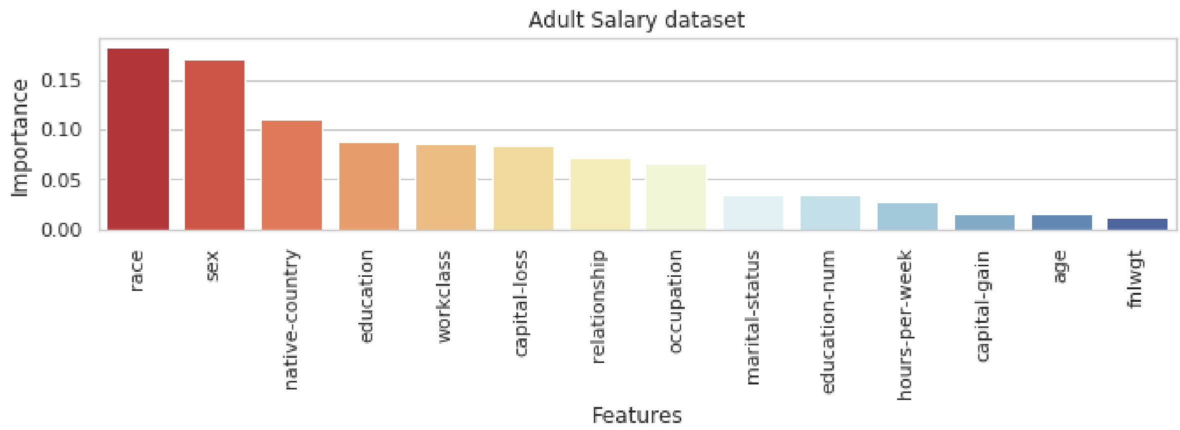


Figure 44. Top in green are the baseline features, bottom red to blue are the better features

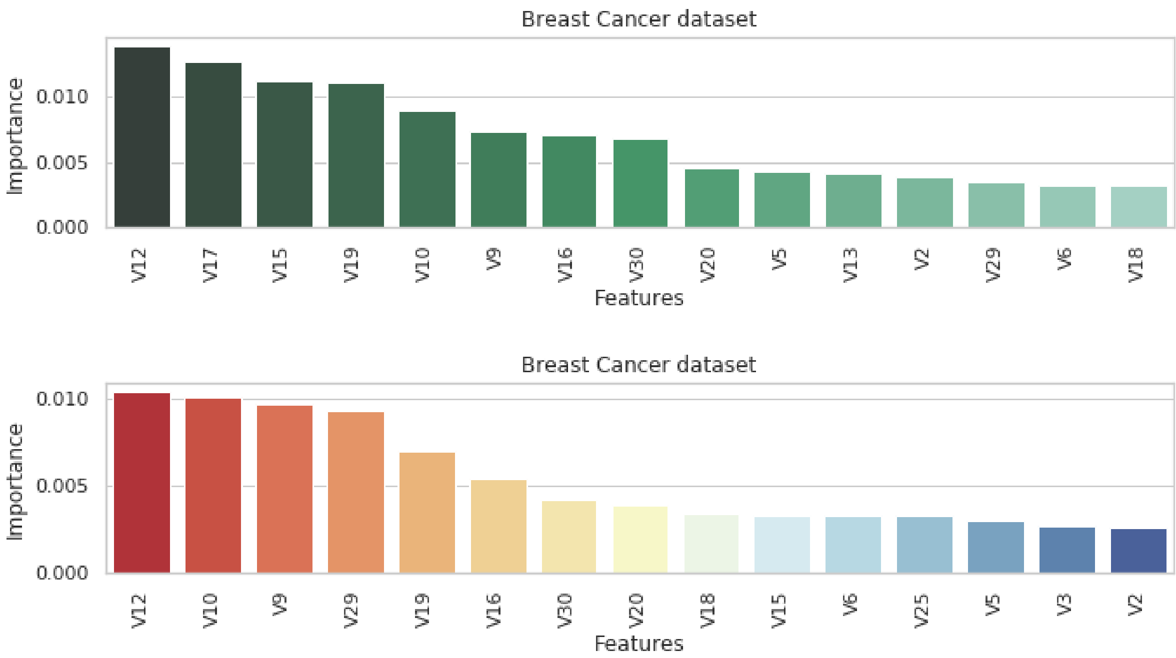


Figure 45. Top in green are the baseline features, bottom red to blue are the better features

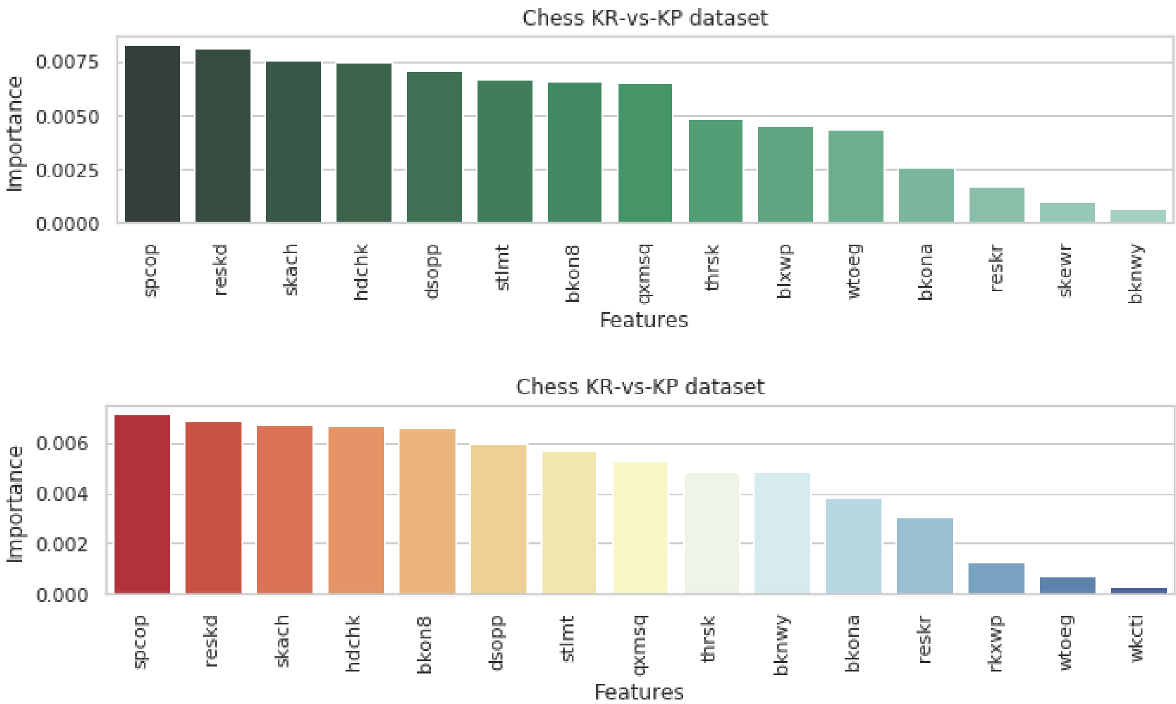


Figure 46. Top in green are the baseline features, bottom red to blue are the better features

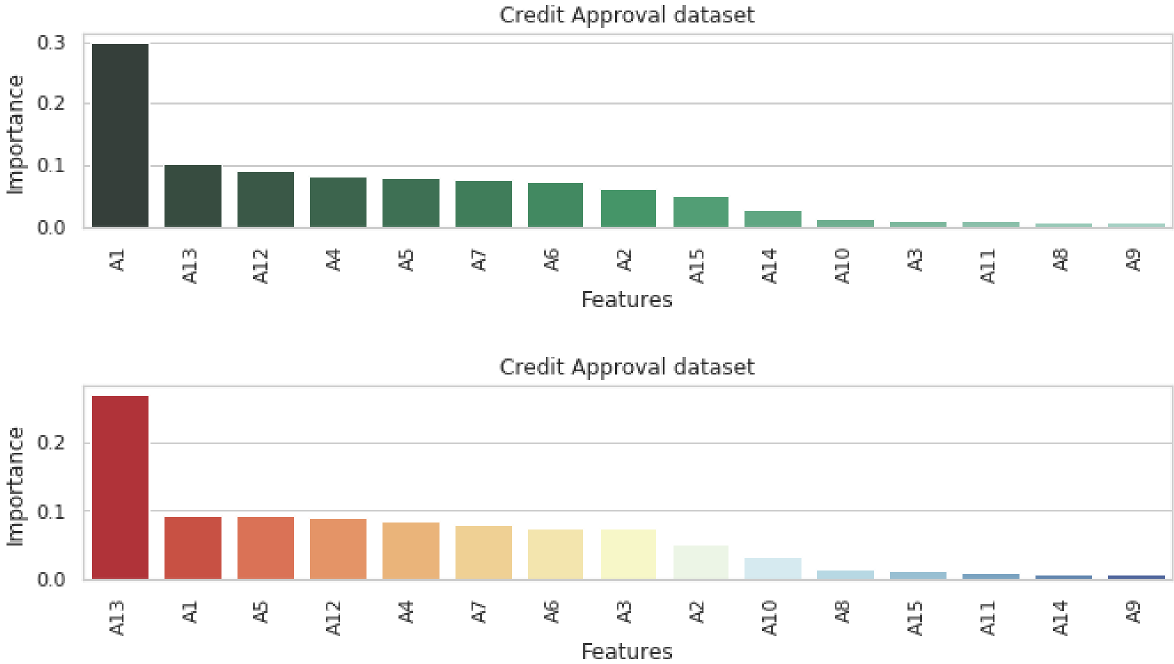


Figure 47. Top in green are the baseline features, bottom red to blue are the better features

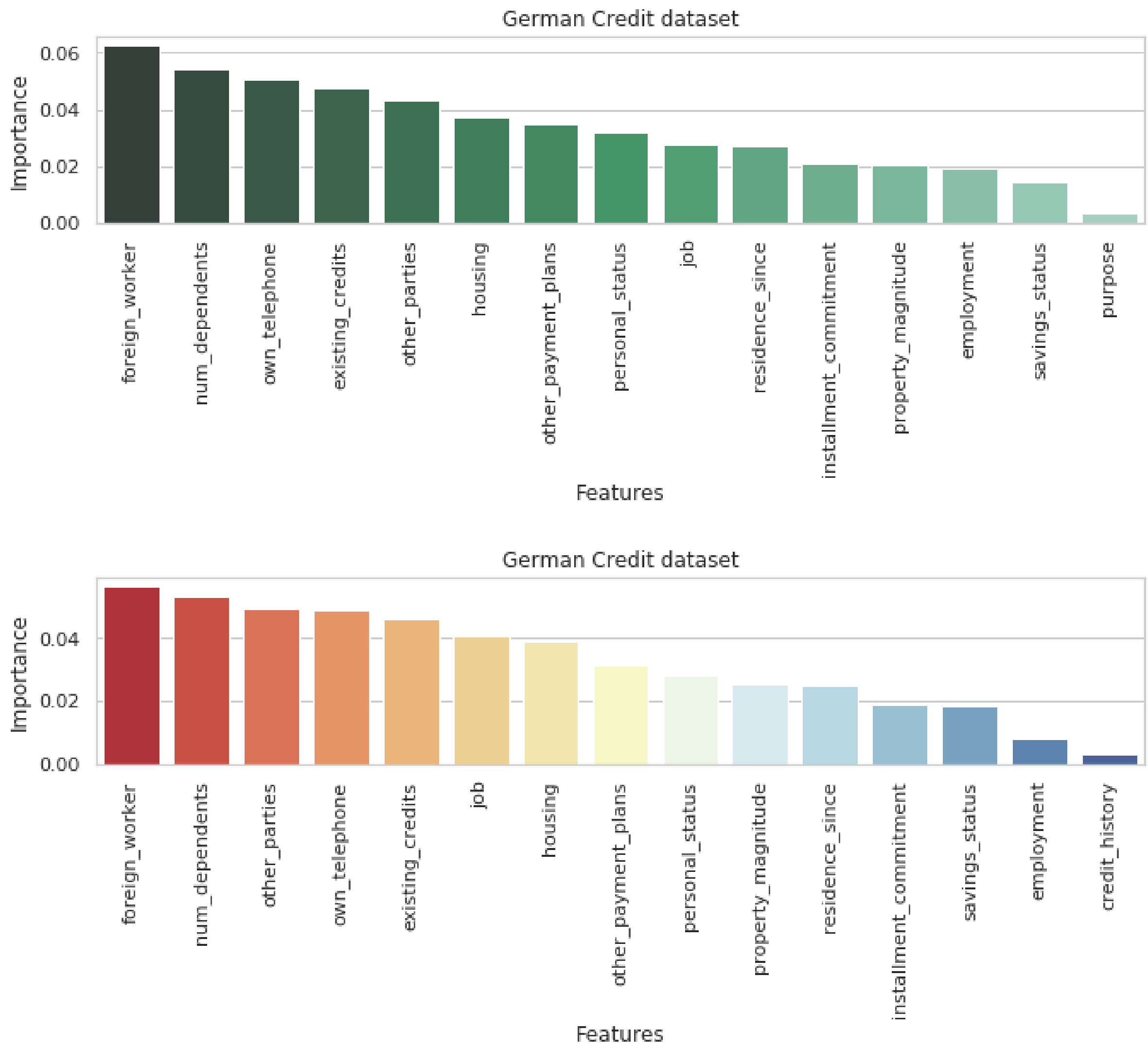


Figure 48. Top in green are the baseline features, bottom red to blue are the better features

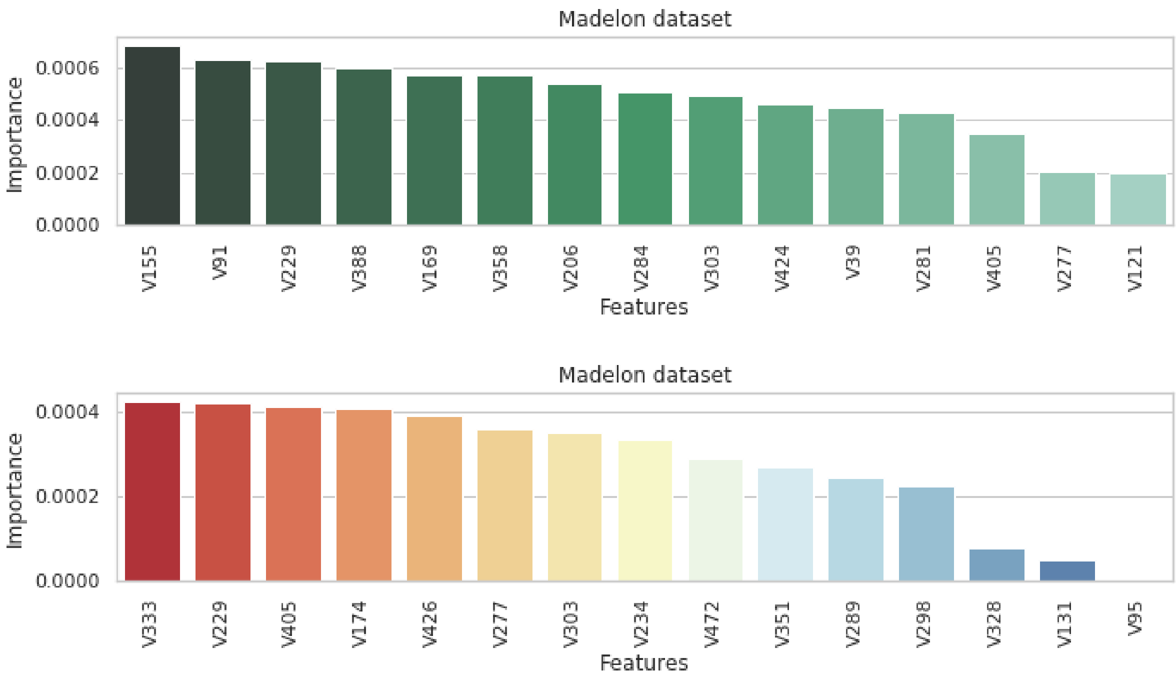


Figure 49. Top in green are the baseline features, bottom red to blue are the better features

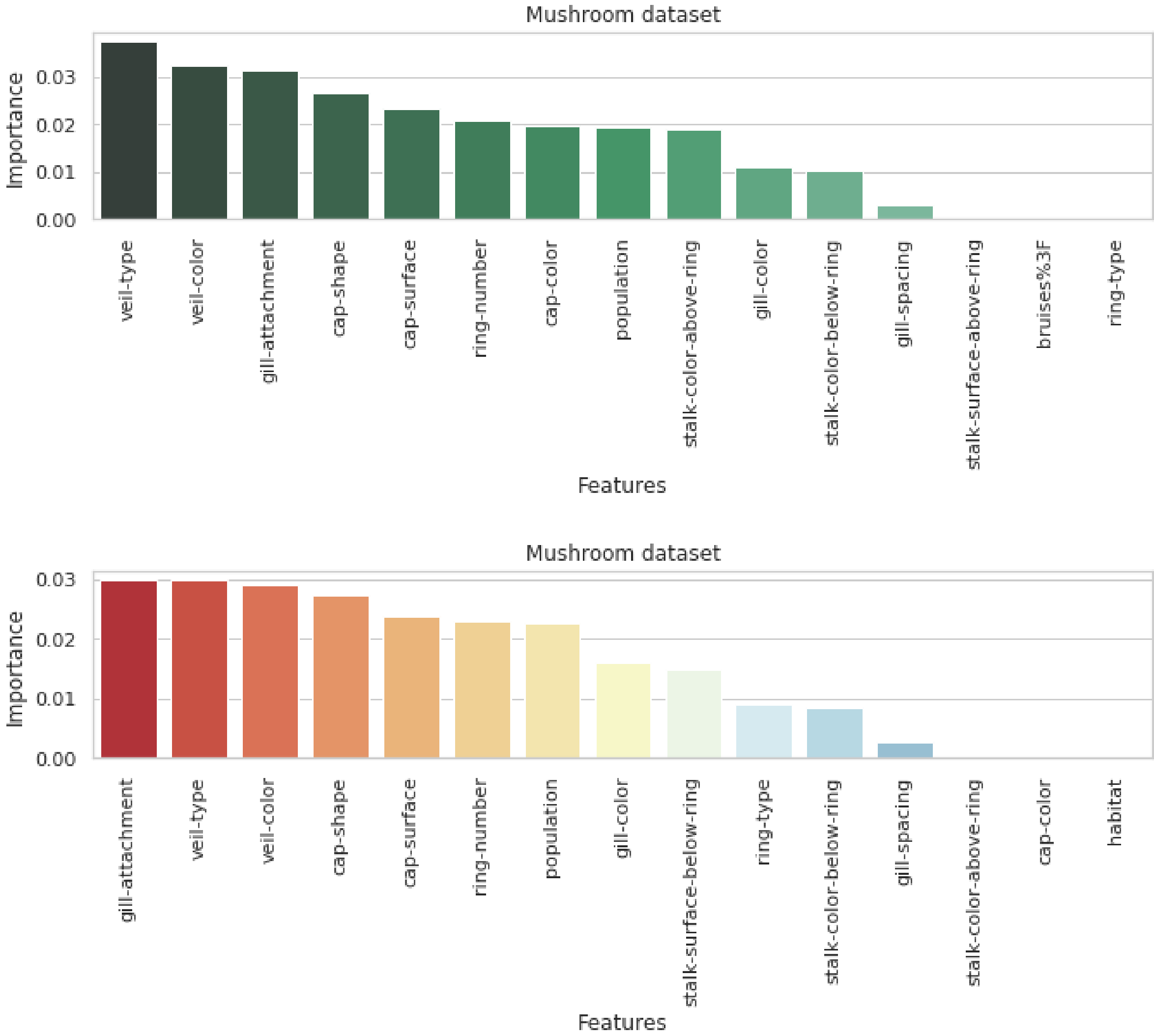


Figure 50. Top in green are the baseline features, bottom red to blue are the better features

Feature Importance of each multi-class dataset, baseline figures are in Green and after applying the proposed improved feature selection approach are in Red to Blue, respectively.

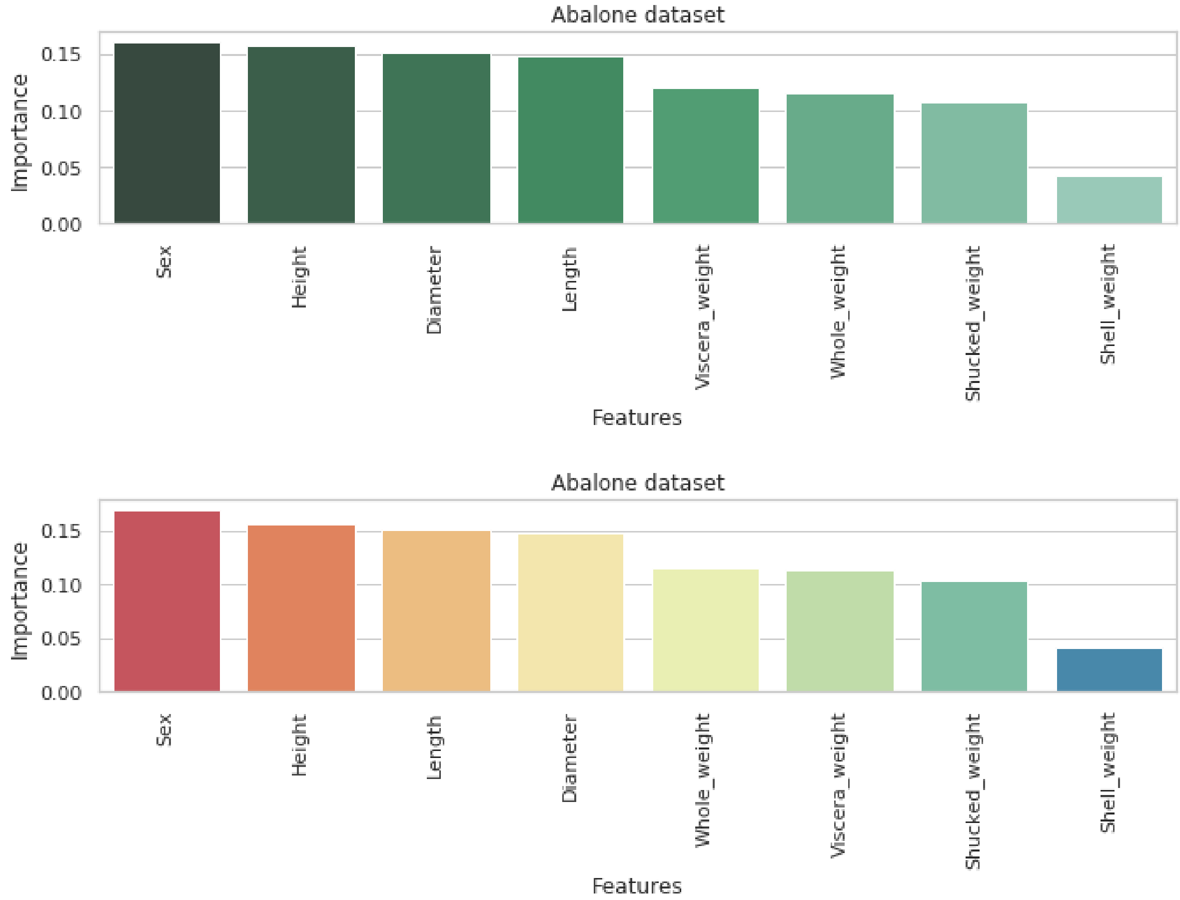


Figure 51. Top in green are the baseline features, bottom red to blue are the better features

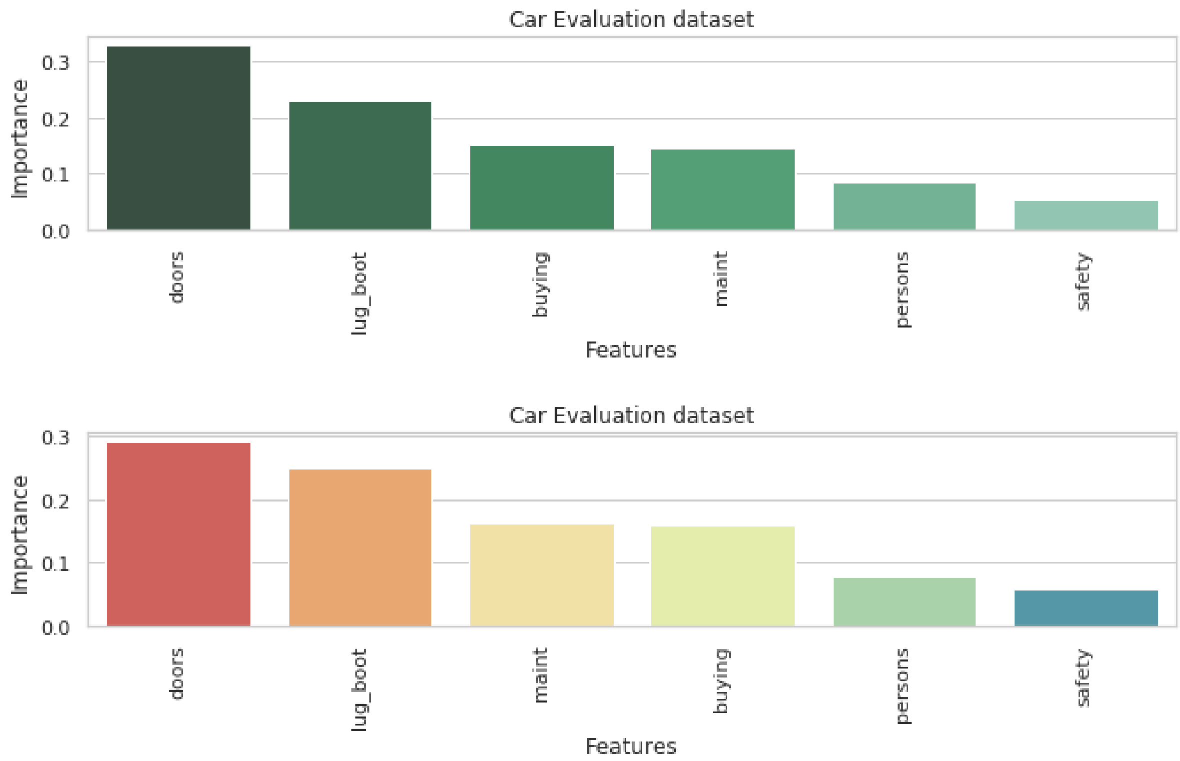


Figure 52. Top in green are the baseline features, bottom red to blue are the better features

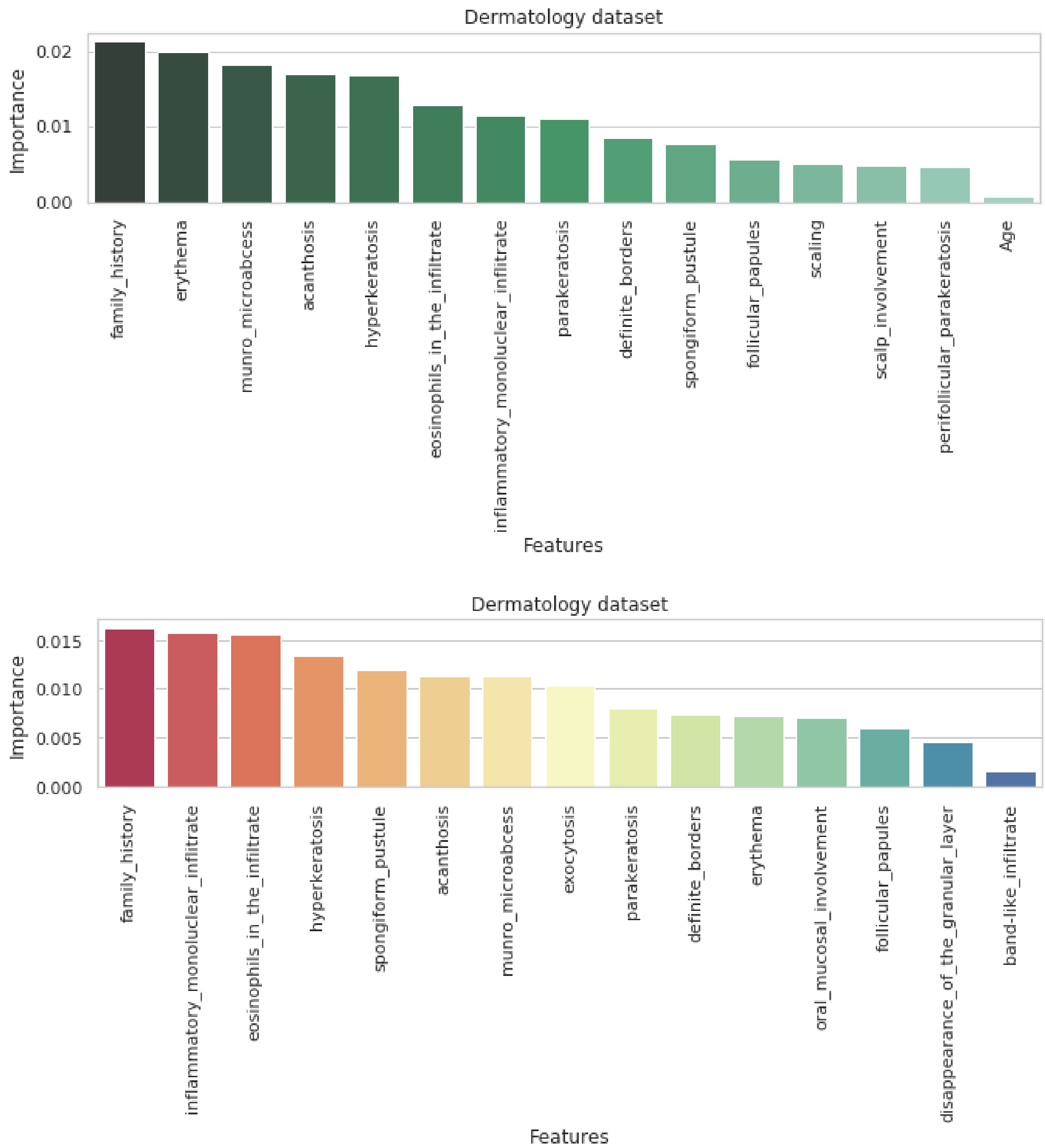


Figure 53. Top in green are the baseline features, bottom red to blue are the better features

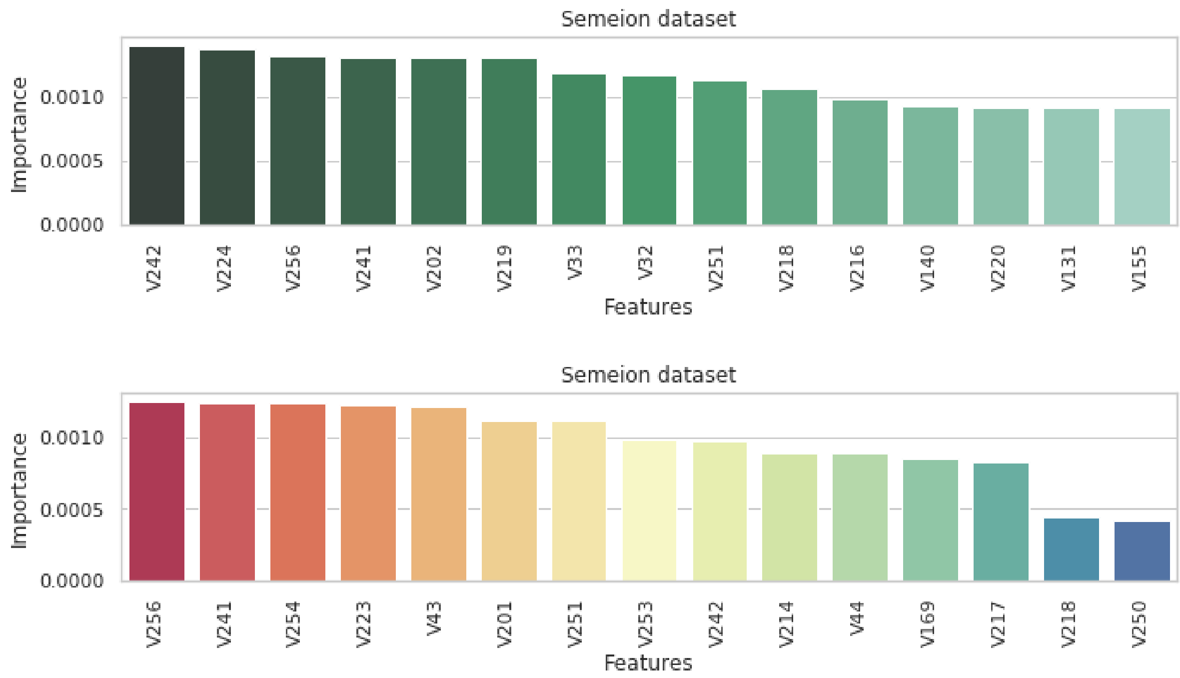


Figure 54. Top in green are the baseline features, bottom red to blue are the better features

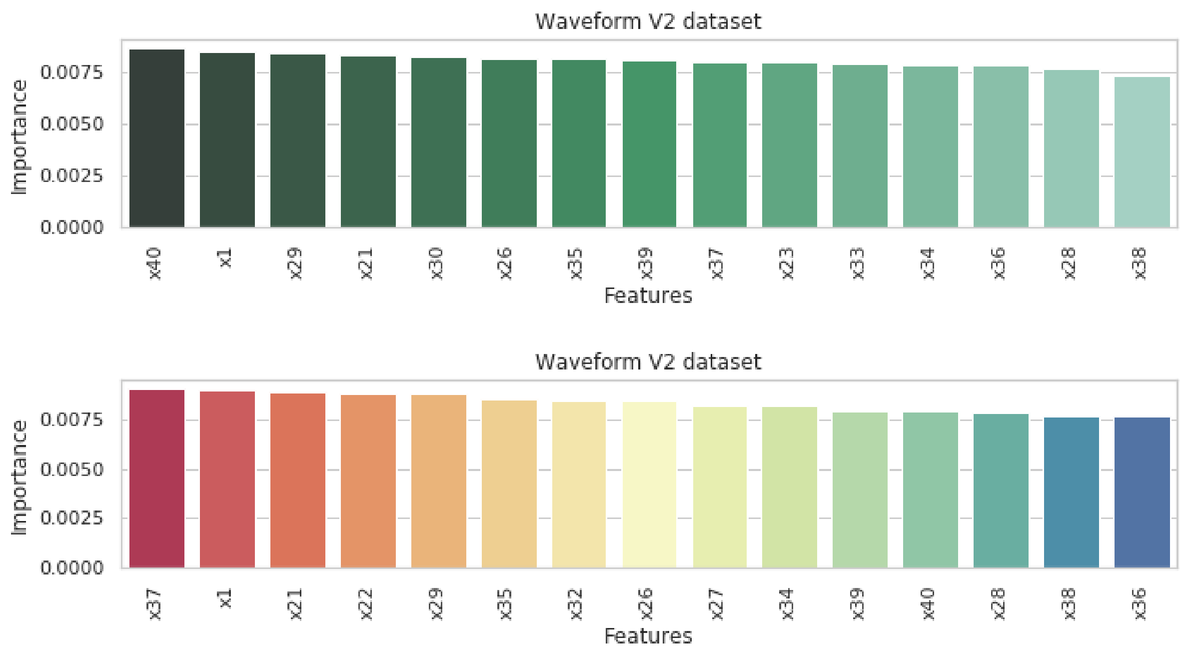


Figure 55. Top in green are the baseline features, bottom red to blue are the better features

Figure 56. Top in green are the baseline features, bottom red to blue are the better features

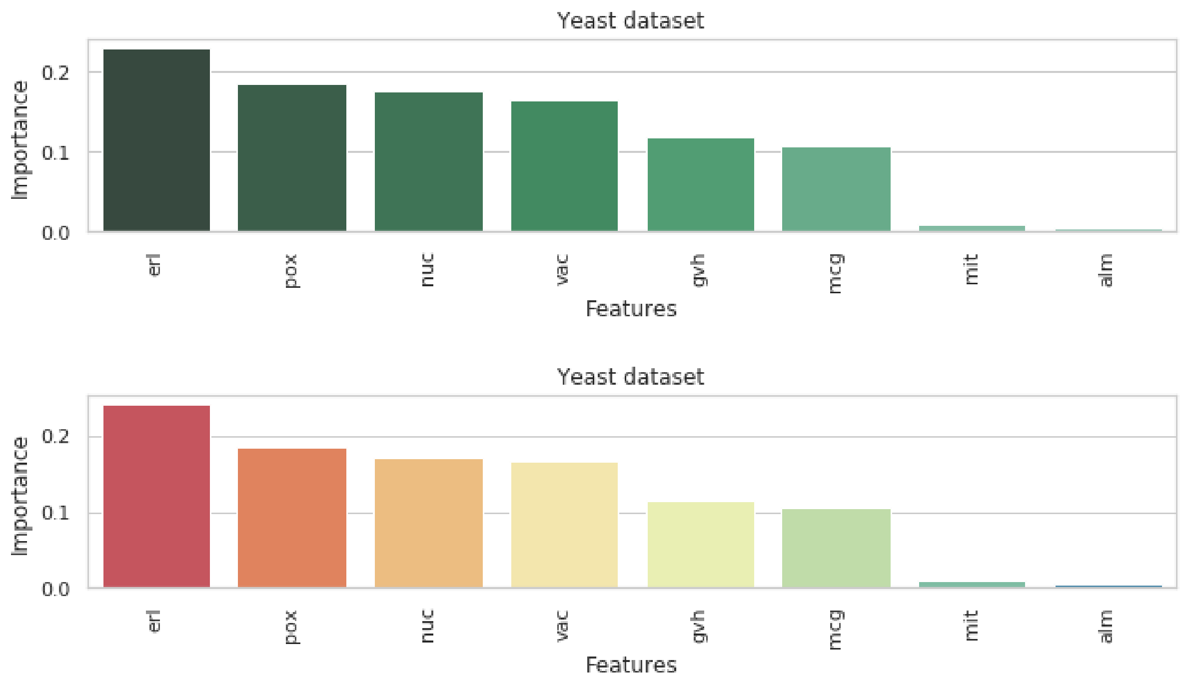


Figure 57. Top in green are the baseline features, bottom red to blue are the better features

Feature Importance of each regression dataset, baseline figures are in Green and after applying the proposed improved feature selection approach are in Red to Blue, respectively.

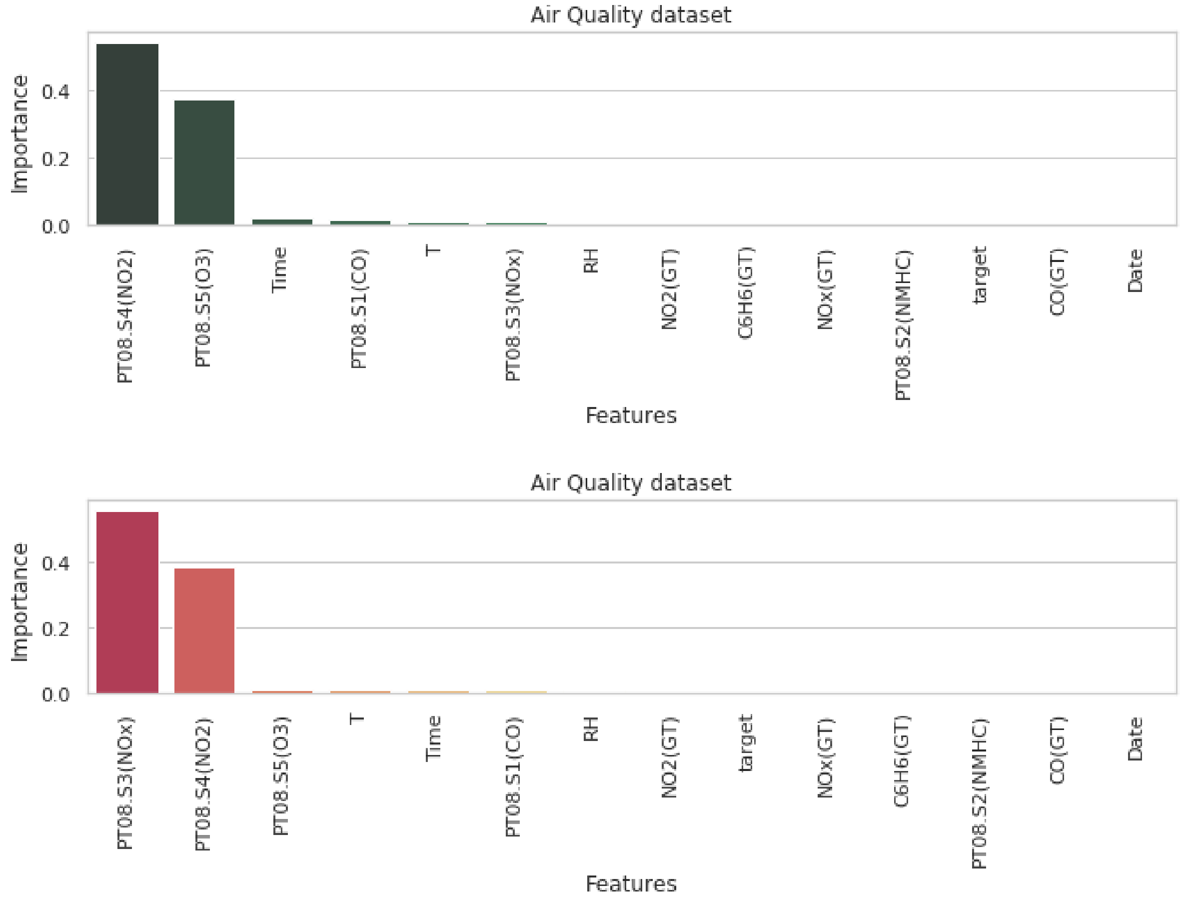


Figure 58. Top in green are the baseline features, bottom red to blue are the better features

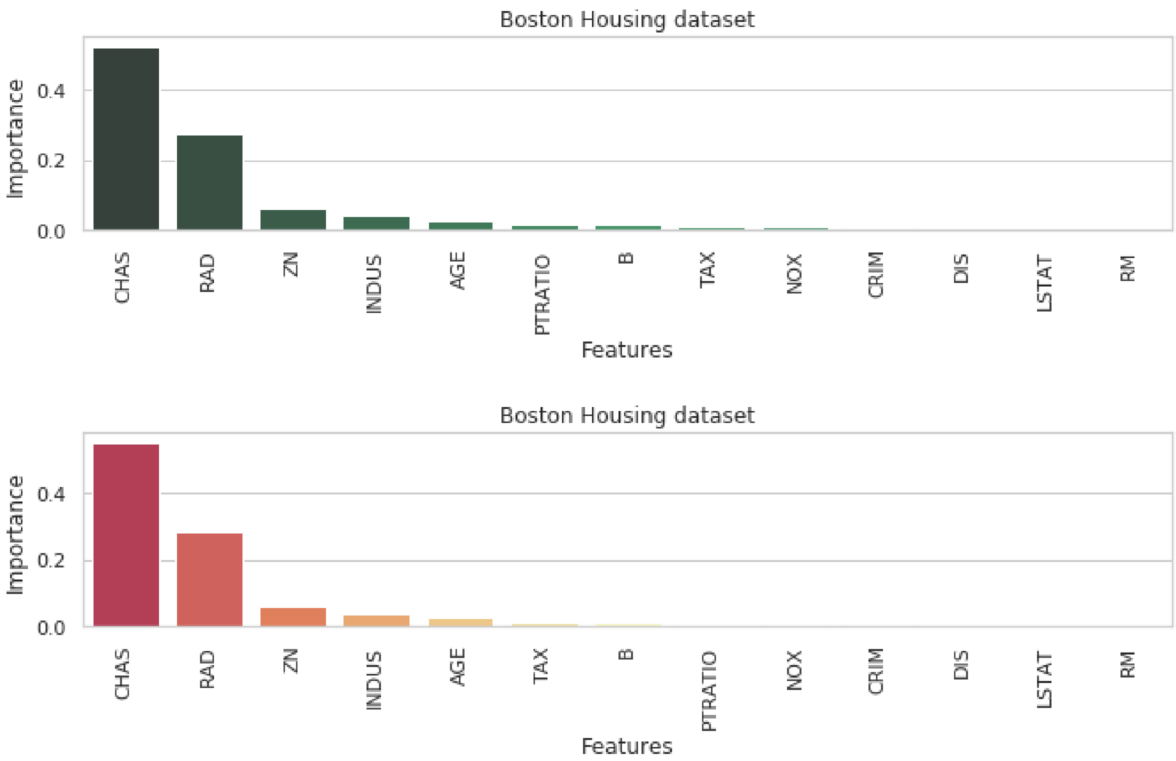


Figure 59. Top in green are the baseline features, bottom red to blue are the better features

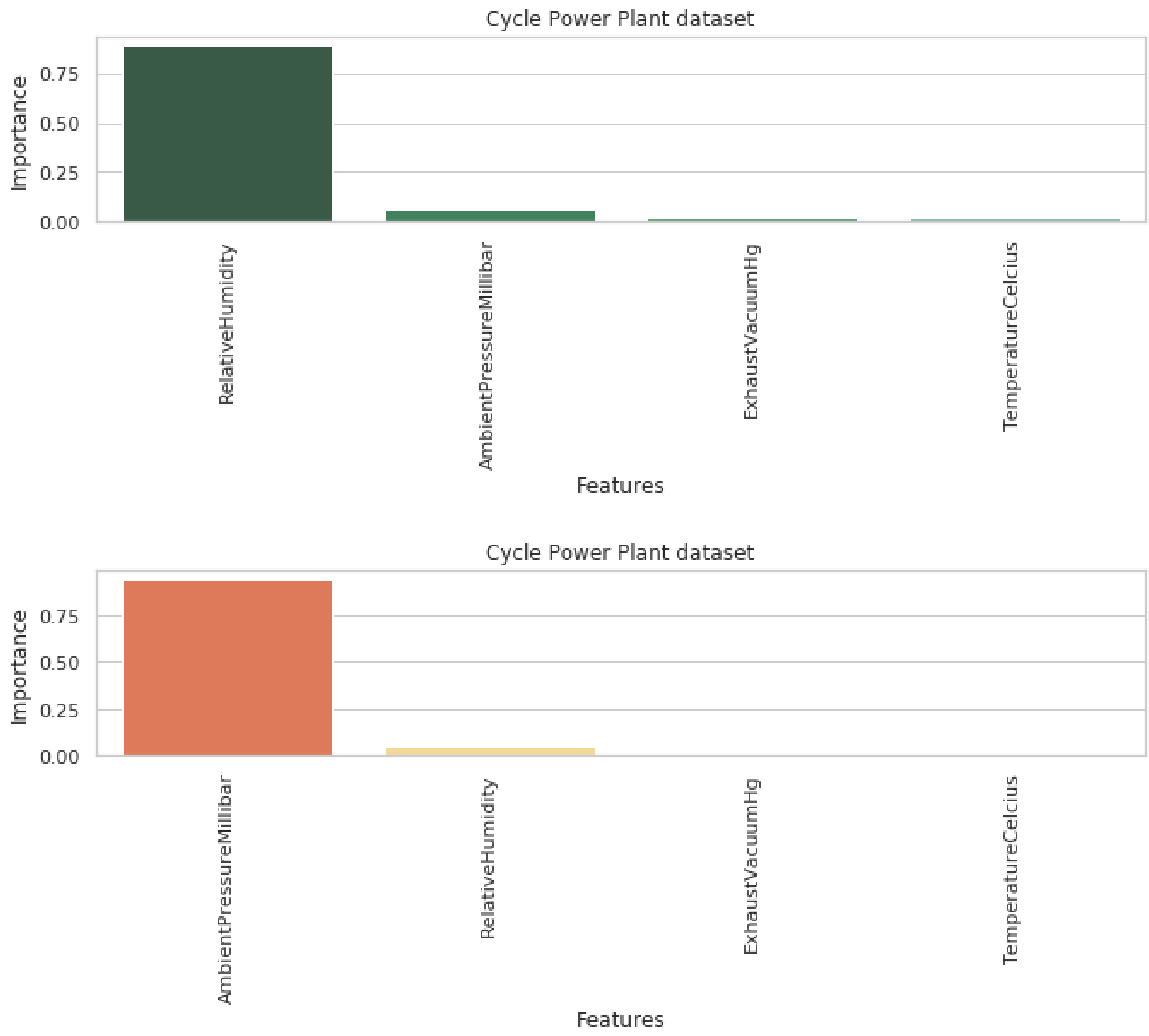


Figure 60. Top in green are the baseline features, bottom red to blue are the better features

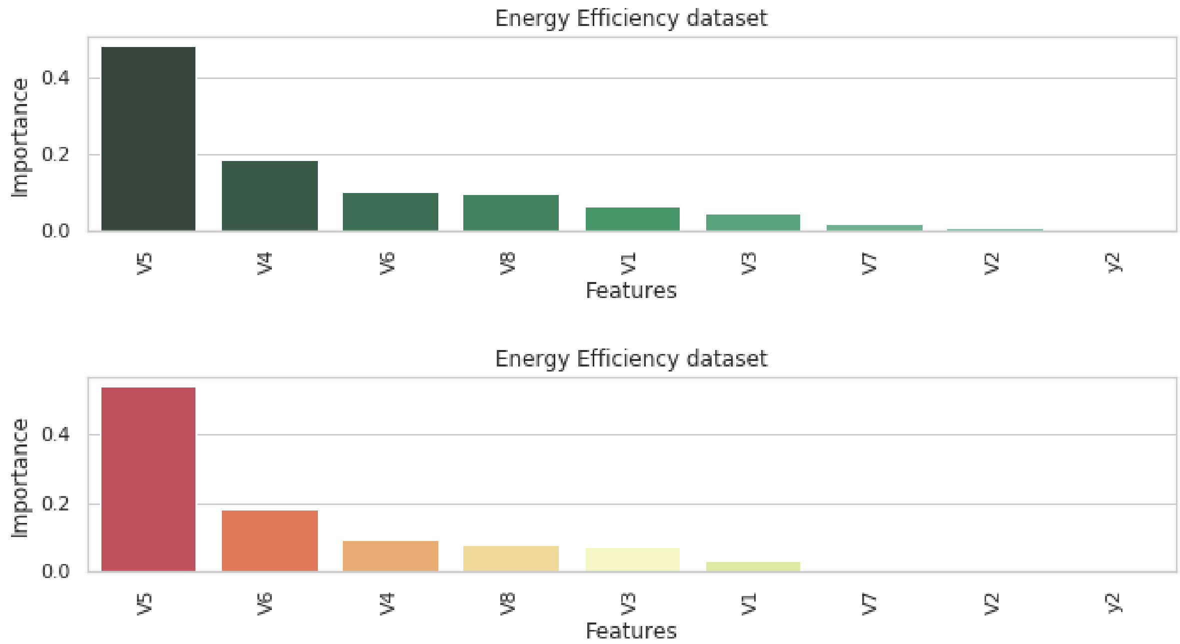
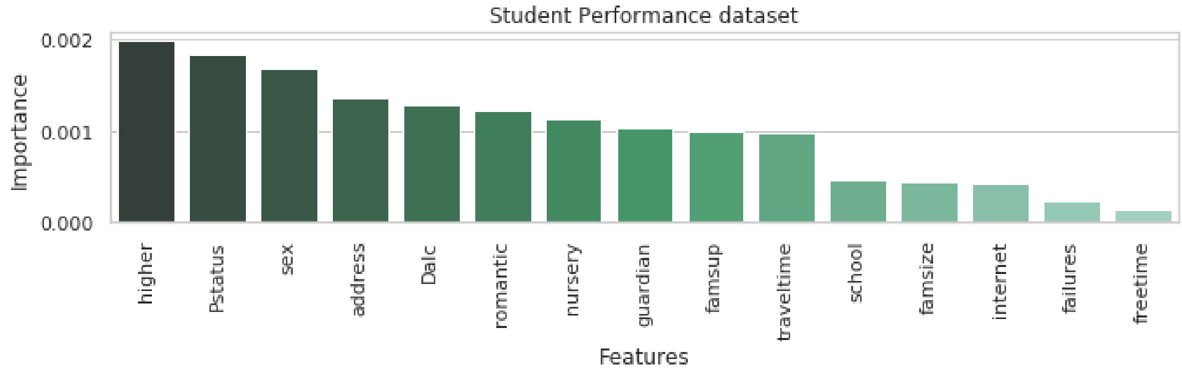


Figure 61. Top in green are the baseline features, bottom red to blue are the better features



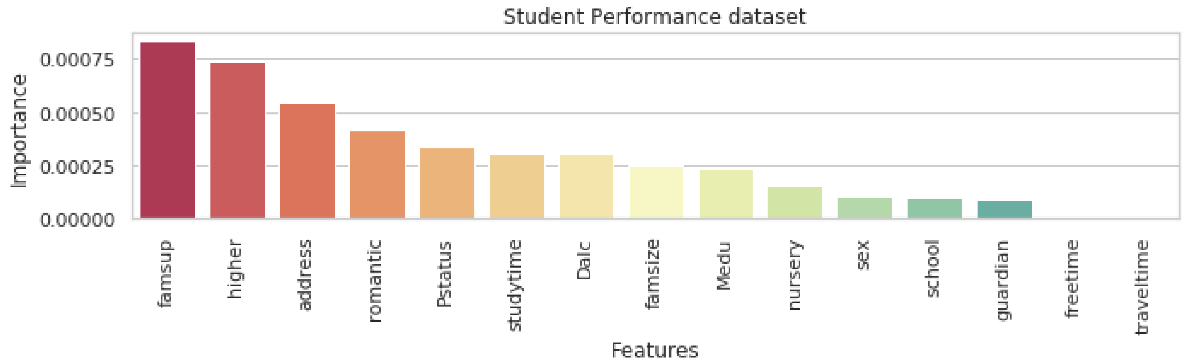


Figure 62. Top in green are the baseline features, bottom red to blue are the better features

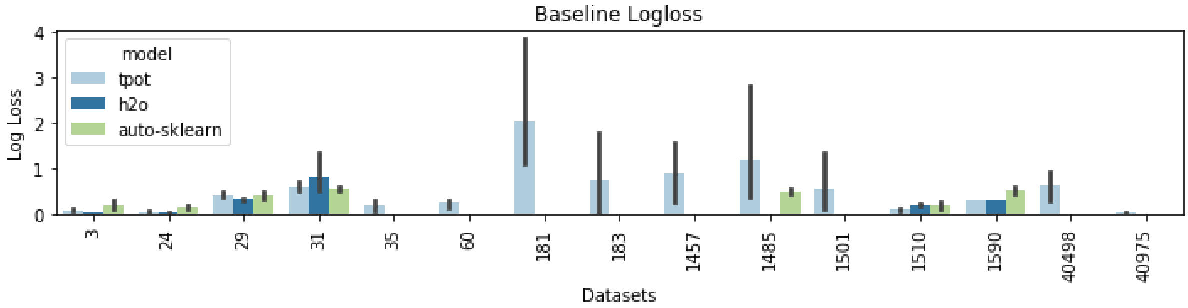


Figure 63. The baseline variance scores for 10 runs each.

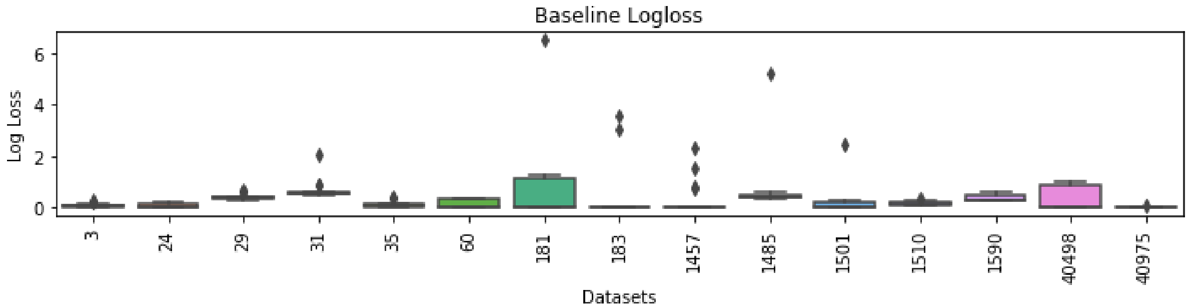


Figure 64. The baseline variance box plot shows a few outliers

Appendix 2:

Figure 65. The low-powered raspberry pi3 used in the experiments

Figure 66. The wearable device running the novel approach