Expert Systems With Applications 247 (2024) 123306

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Prediction of flight delay using deep operator network with gradient-mayfly optimisation algorithm

Desmond Bala Bisandu [a,b,1,*], Irene Moulitsas [a,b,2,*]

[a] Artificial Intelligence and Scientific Computing Laboratory, Department of Computational Engineering Sciences, Cranfield University, Bedford MK43 0AL, UK
[b] Machine Learning and Data Analytics Laboratory, Digital Aviation Research and Technology Centre (DARTeC), MK43 0AL, UK

ARTICLE INFO

Keywords:
Flight delay prediction
Box-cox transformation
Deep residual network
Feature fusion
Deep operator network

ABSTRACT

Accurate flight delay prediction is fundamental to establishing an efficient airline business. It is considered one of the most critical intelligent aviation systems components. Recently, flight delay has been a significant cause that deprives airlines of good performance. Hence, airlines must accurately forecast flight delays and comprehend their sources to have excellent passenger experiences, increase income and minimise unwanted revenue loss. In this paper, we developed a novel approach that is an optimisation-driven deep learning model for predicting flight delays by extending a state-of-the-art method, DeepONet. We utilise the Box-Cox transformation for data conversion with a minimal error rate. Also, we employed a deep residual network for the feature fusion before training our model. Furthermore, this research uses flight on-time data for flight delay prediction. To validate our proposed model, we conducted a numerical study using the US Bureau of Transportation of Statistics. Also, we predict the flight delay by selecting the optimum weights using the novel DeepONet with the Gradient Mayfly Optimisation Algorithm (GMOA). Our experiment results show that the proposed GMOA-based DeepONet outperformed the existing methods with a Root Mean Square Error of 0.0765, Mean Square Error of 0.0058, Mean Absolute Error of 0.0049 and Mean Absolute Percent Error of 0.0043, respectively. When we apply 4-fold cross-validation, the proposed GMOA-based DeepONet outperformed the existing methods with minimal standard error. These results also show the importance of optimisation algorithms in deciding the optimal weight to improve the model performance. The efficacy of our proposed approach in predicting flight delays with minimal errors well define from all the evaluation metrics. Also, utilising the prediction outcome of our robust model to release information about the delayed flight in advance from the aviation decision systems can effectively alleviate the passengers' nervousness.

1. Introduction

The performance of air transportation operation systems has been affected by flight delays leading to substantial economic losses and adversely affecting the planning of passengers, airlines and airports (Guo, 2020; Bubalo and Daduna, 2011; Wang et al., 2011). Flight delay is recognised when discrepancies exist between an aircraft's schedule or arrival and actual departure and arrival time (Zou and Hansen, 2012; He et al., 2022; Chen and Li, 2019). Flights delay contributes to the advantages and disadvantages of airlines, airports and passengers and is unavoidable (Yang et al., 2020; Yu et al., 2019; Lan, 2003). In 2013 31.1 % of flights in the United States were reported to be delayed by 15 min. Also, 36 % of European flights were delayed by 5 min, and 16.3 % of Brazil flights were delayed by 30 min (Sugara and Purwitasari, 2022) These reports show the importance of the flight delay topic and its effects (He et al., 2022; Chen and Li, 2019; Yang et al., 2020; Yu et al., 2019; Lan, 2003; Sugara and Purwitasari, 2022; Mokhtarimousavi and Mehrabi, 2022). The estimated sum of $7.4 million is the annual cost in China caused by the flight delay (Sugara and Purwitasari, 2022). Flight delays in 2007 have a cost implication estimated at $ 32.9 in the economy of the United States of America (USA) (Liu, 2023; Cao et al., 2019). The practical impact of flight delays can affect the passenger experience and performance of the airport. Passengers react with negative emotional dominance, especially service failures (Sugara and

* Corresponding authors.
E-mail addresses: desmond.bisandu@cranfield.ac.uk (D. Bala Bisandu), i.moulitsas@cranfield.ac.uk (I. Moulitsas).
[1] 0000-0002-5723-8429.
[2] 0000-0003-0947-9495.

https://doi.org/10.1016/j.eswa.2024.123306
Received 29 August 2023; Received in revised form 9 November 2023; Accepted 20 January 2024
Available online 31 January 2024

Purwitasari, 2022; Bisandu and Moulitsas, 2023; Bisandu et al., 2023; Bisandu and Moulitsas, 2022). Some causes of flight delays are security, equipment problems and fight crew delays (Sugara and Purwitasari, 2022; Liu, 2023). The delay affected the trade as improved transport relies on customers' trust to minimise or increase ticket sales, so that time flight led to customer self-reliance (Bubalo and Daduna, 2011; Wang et al., 2011; Zou and Hansen, 2012). Thus, flight prediction can lead to an expert decision and functioning for airports and agencies and gratify the client (Bisandu and Moulitsas, 2022; Bisandu et al., 2021; Chen et al., 2023; Tang et al., 2023; Chakrabarty, 2019; Ahmed, 2018; Bisandu et al., 2022; Manasa and Velayutham, 2023; Khan et al., 2023; Faiza and Khalil, 2023; PeCoy and Redmond, 2022; Fatima et al., 2023; Abdelghany et al., 2023). Recently, flight delay has become a severe issue for several travellers and the world's civil aviation industries. To solve the problems of flight delays, the Civil Aviation Administration has devised several programs for reducing the average flight delay time to improve the effectiveness of flight operations. The significant aspect refers to the airport capacity and airspace being inadequate. Several reasons, like airport scheduling, company plans, and luggage, caused delays (Chen and Li, 2019; Yang et al., 2020; Yu et al., 2019).

Despite comprehending the spatial dependence amongst airports on a flight delay, the previous techniques focused on evaluating delays' qualitative impact and are unsuitable for forecasts (Lan, 2003; Sugara and Purwitasari, 2022; Mokhtarimousavi and Mehrabi, 2022; Liu, 2023; Ding, 2017). Even though some of the analyses were produced to predict flight delay, they did not meet the needs of the realistic application. There exist two major issues in prior techniques. The classical techniques consider small sample sizes, so enhancing the prediction accuracy is complex. It is essential to minimise airport dimensions as it can cause data loss (Cao et al., 2019; D. B. Bisandu and I. Moulitsas, "A Deep BiLSTM Machine Learning method for flight delay prediction classification," Journal of Aviation/Aerospace Education Research, vol. 32, no. 2, Jan., 2023; Bisandu et al., 2023). Several sources activate the flight delay, impacting the airlines, airports and on-the-way paths. Some airport techniques combine destination and airlines to analyse the data to forecast flight delays. The airport poses the top precedence of examination on the efficiency using delays produced by carriers. The link between delay and congestion is inspected in an ensemble of routes and airports. Other airlines and airports should be adapted for computing capacity issues and deciding the airlines. Several ways exist to evaluate the complete scope, and it is essential to adapt the air transport models, particularly if the delays in the root are troubled (Bisandu and Moulitsas, 2022; Bisandu et al., 2021; Chakrabarty, 2019).

Various machine learning models are devised in prior works to predict flight delays. Most studies predicted flight delays using binary classifiers, multi-class and regression classifiers. The binary classifier reveals if the flight is delayed or not delayed. The multi-class classifiers produce several delay classes. The regression classifier estimates the value of delay (Zou and Hansen, 2012; Ahmed, 2018; Bisandu et al., 2022). Most of the research is proposed to address the issue of predicting flight delays with classical machine learning methods. Classical machine learning techniques, like decision trees, Bayesian networks and stochastic forests, are introduced for predicting flight delay (Zou and Hansen, 2012; Cao et al., 2019; Manasa and Velayutham, 2023; Khan et al., 2023). Deep models are developed to capture spatial and temporal dependencies to predict flight delays. Predicting precise traffic has acquired focus in managing air transportation because of several advantages. Here, a graph convolutional neural network (GCN) poses the ability to mine complicated nonlinear relations in graphs, bringing prospects in managing complex traffic prediction issues using the deliberation of graph-structured data. The authors have devised intellectual techniques based on GCN models to offer quantified transportation diagnostics (Faiza and Khalil, 2023; PeCoy and Redmond, 2022; Fatima, et al., 2023; Chen et al., 2023). In (Yu et al., 2017), a recurrent neural network is designed for modelling the sequential data to predict traffic. In (Pan et al., 2019), a Meta recurrent neural network is integrated with an encoder-decoder model for predicting urban traffic. In (Zhang et al., 2018), improved GCN is devised by adapting the attention technique for collecting neighbours' features and developing a deep model for predicting the traffic.

This research studies existing airline flight delay prediction approaches and an extant state-of-the-art DeepONet approach proposed by Lu (Lu et al., 2021) for solving nonlinear problems not just in Engineering. This work adds new contributions and novelty to the existing research in the following perspective. We proposed an optimisation-aware deep learning model for predicting flight delays, building upon the state-of-the-art DeepONet method. In this advancement, we carefully consider the weights and biases within the Deep Residual Network (DRN), employing a feature fusion approach that carefully selects an optimal coefficient value. The update mechanism for weights and biases in the DRN is achieved through the Political Social Ski Driver Conditional Autoregressive (PSSDCA) technique, an integration of social ski driver (SSD), CAViaR, and Political Optimisation (PO). Furthermore, we enhance flight delay prediction using the proposed Gradient Mayfly Optimisation Algorithm (GMOA) within the DeepONet framework. Our solution addresses the weight update challenge in DeepONet using GMOA, a blend of Mayfly Algorithms (MA) and the gradient descent algorithm, showcasing efficacy in the improved flight delay predictive task performance with minimal error. This approach is distinctive from all existing approaches because it performs weight and biases using optimal global solutions from the optimisation algorithm proposed to enhance the DeepONet performance in terms of input quality, which monitored and improved the training and prediction phases of the techniques to avoid model overfitting.

The major contributions of the paper include:

- **Proposed PSSDCA-based DRN for fusing features:** The fusion of features is performed with a designed PSSDCA-based DRN. The features are employed on DRN to perform feature fusion with weights and biases based on hidden layers. The DRN training is performed using the proposed PSSDCA and obtained by combining SSD, CAViaR and PO by tuning optimum weights.
- **Proposed GMOA-based DeepONet to predict flight delay:** The prediction of flight delay is performed using the proposed GMOA-based DeepONet. The weight update of DeepONet is achieved using the proposed GMOA by training the DeepONet. The MA is integrated with a gradient descent algorithm for selecting the best weights by minimising the error.

The rest of the sections are as follows: Section 2 discusses the literature review on flight delay prediction models. Section 3 describes the proposed model for the flight delay. Section 4 contains the results and discussions, while Section 5 presents conclusion and lessons learned.

2. Literature review

Several studies about airport flight delay prediction and management exist (Cai et al., 2015; Chen et al., 2009; Zhang et al., 2019; Qu et al., 2020; Liu et al., 2018; Huo et al., 2020; Zoutendijk and Mitici, 2021; Dutrieux, 2021). The flight delay prediction consists of arrival or departure time, depending on the interest of the study. The aim of the prediction is generally to predict the flight arrival or departure time accurately. The results of the predictions are utilised for decision-making. Thus, the accuracy of the prediction affects the entire system. Several classical forecasting methodologies and prediction theories have been mentioned in the literature, including the Naïve Bayes model, Support Vector Machine, Logistic Regression, multivariate regression and time series prediction (Ma et al., 2023). However, these methods and theories may be less suitable as flight delays is a multidimensional time series with features of high complexity that need more sophisticated fusion procedure (Ben Messaoud, 2021; Karádi et al., 2015; Jiang and Zhang, 2016). Therefore, we introduce a novel approach based on the state-of-the-art approach called DeepONet (Lu et al., 2021) integrated with an optimisation algorithm to improve the training and prediction performance of the model by reducing the error to the barest minimum. In particular, the input is a function *f(x),* including all the important factors that lead to delays. Predicting flight delays is easier based on the dynamics of the flight and neural network, together with the real-world data collected from the BTS. Additionally, the model performance is improved by the pre-processing procedure proposed.

The eight conventional flight delay prediction methods and their merits and demerits are enlisted in Güvercin M et al., (Guvercin et al., 2020), the authors devised clustered airport modelling techniques to predict the delays in flight considering airport networks. The technique offered a precise prediction of flight delays. However, the method did not choose imperative features to forecast delay. Lambelho Met al., (Ma et al., 2023) devised a generic analysis of strategic schedules at an airport with a flight delay to include imperative features. The method offered good efficiency, but this technique did not provide improved prediction accuracy. To enhance accuracy, Chen Zet al., (Chen et al., 2017) developed the Information Gain-Support Vector Machine method for discovering the Chinese airline efficiency under Carbon dioxide (CO2) emissions and flight delays. The method minimised the issues of the classical Data Envelopment Analysis (DEA) model but did not offer more policy implications. Chen J and Li M (Chen and Li, 2019) devised a Machine learning technique for predicting flight delays to offer policy implications. The method offered a precise model for delay prediction but did not involve an effectual database to enhance the accuracy.

To handle another database, Alla et al. (He et al., 2022) utilised artificial neural networks (ANN) and Multilayer perceptron (MLP) for predicting flight delay. Here, the authors extract chronological flight data to enhance efficiency. The technique offered better accuracy but was unsuitable for real-time flight data. To deal with real-time data, Yi, Jet al., (Yi et al., 2021) developed a standard Stacking classification technique for predicting flight delay. The method effectively dealt with the imbalanced datasets, and the Boruta algorithm was used to choose effectual features. However, the method did not include other machine learning methods to learn flight delays. To include machine learning methods, Guo, et al., (Guo et al., 2021) devised a hybrid model, Random Forest Regression and Maximal Information Coefficient (RFR-MIC), for predicting flight delay. The Regression Maximal Information Coefficient and Random Forest were combined to predict the delay. However, the technique did not use cloud or parallel computing to minimise the computation time. To minimise the computation time, Shao, Wet al., (Shao et al., 2022) devised a vision-based solution for predicting flight delay. A deep end-to-end model, TrajCNN, was utilised to capture spatial and temporal data using the situational awareness map. However, the method was not applicable for long-term prediction.

Bisandu et al., (Bisandu et al., 2022) developed a novel method for an accurate flight forecast using social ski driver and conditional autoregressive value at risk. Their proposed model outperforms existing models and *meta*-heuristic approaches in predicting flight delays. However, the authors recommend other approaches to be developed based on other optimisation algorithms compared with their proposed models' performance. In (Bisandu et al., 2021); Bisandu et al., design a model for predicting flight delay using a deep feedforward network. Their model's performance is compared to shallow architectures, such as support vector machines and single-layer neural networks. Their model outperforms the shallow architectures regarding the accuracy and classification of flight delays. Kim et al., (Ayoubi, 2018) proposed different architectural designs and implementation of LSTM and RNN in predicting flight delays using sequences of thresholds. Lin et al., (Cai et al., 2017) the authors performed airport flight delay prediction using convolutional LSTM (Conv-LSTM) using temporal and spatial characteristics in China civil aviation. Their research shows the importance of parameter tuning when dealing with huge datasets and deep-learning neural networks.

The authors in (Divya et al., 2023) conducted a study that focused on addressing the challenges associated with flight planning in the business world, especially concerning unpredictability and flight delays. Flight delays incur significant financial losses for airlines, providers, and travellers, making their prediction and analysis crucial. The research aimed to bridge the information gap between airlines and passengers by predicting the possibility of flight delays based on aircraft characteristics. The analysis was centred around scheduled arrival, departure times and actual time. The study extensively evaluated various machine learning techniques for predicting flight delays, particularly on Artificial Neural Networks (ANN). The authors found that ANN, optimised using the Adam Optimization algorithm, exhibited high accuracy in modelling sequential data and proved to be the most effective approach for anticipating flight delays among the algorithms compared. Furthermore, the researchers proposed a novel aviation delay prediction method, which combined ANN with a genetic algorithm (GA) to optimise important parameters for flight delay prediction. The resulting GANN model demonstrated a strong performance, achieving an accuracy score of 89 %, affirming its ability to forecast flight delays accurately. This research offers valuable insights and a potential prototype for identifying operational factors contributing to flight delays in diverse situations.

In the study, the authors in (Tirtha et al.,May, 2023) recognised the critical importance of accurate airport traffic flow estimation for the safe and efficient operation of the aviation system. While recent advancements in machine learning have shown promise in predicting traffic flow at individual airports, these approaches often neglected the intricate spatial interactions among multiple airports, making them less effective when applied to the aviation system comprising numerous airports. To address this gap, the authors developed a novel spatiotemporal hybrid deep learning model tailored for the multi-airport scenario. This model efficiently captured the spatial correlations and temporal dependencies among airports in a parallelised manner. They achieved this by introducing causal inference techniques to model how different airports interact, creating adaptive causality graphs based on data-driven insights to account for the diversity among airports. Moreover, recognising that not all airports could utilise the same data sources for feature extraction, the authors designed a feature mask module to choose relevant features during spatial information mining selectively. The authors conducted extensive experiments using real data from China's top 30 busiest airports. Their results demonstrated that their spatio-temporal deep learning approach outperformed state-of-the-art methodologies by up to 4.7 % against established benchmarks. Ablation studies reinforced the significance of the adaptive causality graph and the feature mask module, further confirming the effectiveness of their proposed methodology. Overall, this research presents a robust framework for enhancing traffic flow prediction in multi-airport scenarios, focusing on modelling spatial interactions and adapting to the unique characteristics of individual airports.

The authors in (Tirtha et al.,May, 2023) factors influencing flight-level airline delays by simultaneously considering both departure and arrival delays. They introduced a novel statistical model system based on copulas, specifically a group generalised ordered logit (GGOL) model, to accomplish this. This model was designed to account for the observable and unobservable effects that influence delays in flight departure and arrival times. The authors analysed the 2019 marketing carrier on-time performance data from the Bureau of Transportation Statistics (BTS), covering 67 airports in the continental United States. They enriched the delay data with a comprehensive set of independent variables, including traffic conditions at the origin and destination airports before flight departure and arrival, trip-related attributes, weather variables spanning the entire flight duration, and spatial and temporal considerations. Their model estimation revealed that the Joe copula model, with a specific parameterisation, provided the best fit to the data. The model's performance was further validated using a holdout sample, demonstrating its excellent predictive capabilities. The authors conducted a prediction exercise under various hypothetical scenarios to demonstrate the model's practicality and utility for prediction. This exercise showcased the model's potential as a valuable tool for predicting airline-carrier-level or airport-level delays while accounting for weather forecasts and a range of independent variables. In summary, the study offers an innovative approach for jointly modelling flight departure and arrival delays, providing insights that could be applied to improve airline operations and enhance delay prediction and management.

The authors in (Faiza and Khalil, 2023) analysed the impact of the COVID-19 pandemic on the on-time performance of the US airline industry during disrupted situations. They employed deep learning techniques, specifically neural network regression, decision forest regression, boosted decision tree regression and multi-class logistic regression, to assess and compare their effectiveness in predicting flight delays. The study aimed to identify the most suitable technique for this purpose and offer data-driven recommendations for improving on-time performance in disrupted situations. Their analysis indicated that all the techniques used yielded satisfactory performance. The coefficient of determination for neural network regression was 0.86, while for decision forest regression, it was 0.85. The boosted decision tree regression achieved a coefficient of determination of 0.870984, making it the best-performing technique. Multi-class logistic regression, on the other hand, exhibited an overall accuracy and precision of 98.4 % and a recalling/remembering performance of 99 %, establishing it as the best model for predicting flight delays during the COVID-19 pandemic. Furthermore, the study explored the relationships between different types of delays and on-time performance. It found a strong positive relationship with departure delay (94 %), carrier delay (53 %), late aircraft delay (35 %), weather delay (21 %), and NAS (National Airspace System) delay (14 %). However, the study revealed a weak negative or almost unrelated relationship between airtime and arrival delay. Security and arrival delays were also found to be almost unrelated, with a minimal 1 % relationship. Based on these findings, the authors recommended taking measures to significantly reduce departure delay, carrier delay, late aircraft delay, weather delay, and NAS delay, which collectively accounted for 94 %, 53 %, 35 %, 21 %, and 14 % of the on-time performance impact in disrupted situations. The proposed models had a Mean Absolute Error (MAE) of 2 % for Neural Network Regression, Decision Forest Regression, and Boosted Decision Tree Regression, and Root Mean Square Error (RMSE) of approximately 11 %, 12 %, and 11 %, respectively, indicating their effectiveness in predicting flight delays in the context of the COVID-19 pandemic.

In (Wu et al., 2023), the authors addressed the critical need for accurate and interpretable delay predictions in the aviation industry. They recognised the complexity of incorporating spatiotemporal dependencies and external factors influencing delay propagation. To overcome this challenge, they introduced the SpatioTemporal Propagation Network (STPN), a novel approach that effectively models delay propagation by considering both spatial and temporal factors. STPN employs a multi-graph convolution model to account for geographic proximity and airline schedules from a spatial perspective. Additionally, it integrates a multi-head self-attention mechanism to capture various types of temporal dependencies in delay time series, ensuring a comprehensive understanding of the temporal aspect of delay propagation. Importantly, this network can learn these features end-to-end, offering a holistic approach to modelling delay dynamics. The authors conducted experiments on real-world delay datasets, demonstrating that STPN surpasses state-of-the-art methods in multi-step ahead arrival and departure delay prediction, particularly in large-scale airport networks. STPN's ability to generate counterfactuals adds to its appeal by providing transparent insights into the explainable patterns of delay propagation. Comprehensive experiments further established STPN as a robust benchmark for general spatiotemporal forecasting, indicating its potential to enhance decision-making within the aviation industry. This research presents a novel and promising approach for improving delay prediction, especially in complex spatiotemporal environments, and offers valuable insights into the dynamics of aviation delays.

(Cai et al., Mar. 2023); In their research, the authors tackled the crucial problem of flight delay prediction, a topic of significant interest in civil aviation due to its essential role in airline planning, flight scheduling, airport operations and passenger service. Many factors influence flight delays and tend to irregularly propagate through air transportation networks, primarily due to flight connectivity, which poses a substantial challenge for accurate prediction. Recent years have seen the rise of Graph Convolutional Networks (GCNs) in flight delay prediction due to their ability to extract complex relationships. However, many existing GCN-based methods have struggled to capture spatial–temporal information crucial for precise prediction effectively. In response, the authors introduced a novel approach known as the Geographical and Operational Graph Convolutional Network (GOGCN) for multi-airport flight delay prediction. GOGCN is a GCN-based spatial–temporal model designed to enhance the representation of node features by considering both geographical and operational spatial–temporal interactions in a graph. It incorporates two essential components: Operational Aggregator: This component extracts global operational information by analysing the graph structure, thus enabling the model to incorporate critical operational aspects that influence flight delays. Secondly, Geographical Aggregator: This element captures the similarities among spatially proximate airports, recognizing that geographically close airports might share similar patterns in their delay propagation. Extensive experiments conducted on a real-world dataset showcased the effectiveness of the proposed GOGCN approach. It outperformed state-of-the-art methods, achieving a significant improvement in prediction accuracy. This research presents an innovative solution that enhances the accuracy of flight delay prediction by addressing the spatial–temporal challenges inherent in the aviation industry, ultimately benefiting airline planning, operations and passenger services.

In (Khan et al., Jan. 2024), the authors addressed the critical task of predicting and analysing flight delays, which is fundamental for efficient air traffic management and control. They introduced a novel parallel-series model and a unique adaptive bidirectional extreme learning machine (AB-ELM) method to enhance flight delay prediction and feature analysis to gain a deeper understanding of the causes of flight delays, as defined by the International Air Transport Association (IATA). A distinguishing feature of their work is the examination of IATA-coded flight delays, a dimension that has been largely overlooked in existing research. Using their proposed parallel-series model, the authors refined the decision boundaries for IATA-coded flight delay subcategories. This model is particularly valuable in scenarios where traditional multiclass-multilabel classification methods might yield erroneous results. The AB-ELM method introduced in the study focuses on improving network generalization performance by adaptively adjusting the learning rate during gradient ascent, as opposed to the more commonly used gradient descent. To validate their approach, the authors conducted a case study using historical data from one of Hong Kong's international airlines. This dataset encompassed a wide range of information, including details about the airport, flight, aircraft, weather conditions, and IATA flight delay subcategories. The study employed fourteen different sampling approaches to mitigate the impact of imbalanced and noisy data. The study's results demonstrated the effectiveness of combining appropriate sampling techniques with the parallel-series model and AB-ELM method in uncovering hidden patterns within the complex system of IATA-coded flight delay subcategories. Notably, the AB-ELM method achieved a high accuracy rate of 80.66 %, outperforming other data-driven approaches. This research provides valuable insights for airlines, enabling them to develop proactive contingency measures based on potential flight delay causes and durations. The study contributes to improved air traffic management and enhances airlines' ability to manage and mitigate the impact of flight delays on their operations.

The authors in (Li et al., Oct. 2023) focused on the crucial field of flight delay prediction, which plays a pivotal role in intelligent aviation systems, influencing flight scheduling, airline planning and airport operations. Accurately predicting flight delays is challenging due to the diverse factors contributing to delays. Furthermore, the aviation system's interconnected nature results in complex spatial–temporal correlations involving spatial relationships between airports and temporal dependencies among timestamps. To address these challenges, the authors introduced a novel deep learning framework, the CNN-LSTM model, designed to consider both spatial–temporal correlations and extrinsic features for flight delay prediction. This model comprises two key components: Convolution Neural Network (CNN): The CNN architecture is employed to capture spatial correlations, enabling the model to understand relationships between different airports in the aviation network. Long Short-Term Memory (LSTM): The LSTM architecture captures temporal correlations, allowing the model to learn patterns and dependencies in flight delays over time. The spatial–temporal correlations derived from the CNN-LSTM framework are combined with extrinsic features, such as airline-specific issues, distance, scheduled flight times, etc. These fused inputs are then used to train a Random Forest (RF) model for flight delay prediction. The authors validated their approach using data from U.S. domestic flights in 2019 from the Bureau of Transportation Statistics. The results demonstrated the impressive accuracy of the CNN-LSTM model, achieving an accuracy rate of 92.39 %. Specifically, the model correctly identified approximately 91 % of on-time flights and achieved an 84 % classification accuracy for delayed flights, outperforming several benchmark models. The prediction model developed in this study has practical implications for airport regulators, offering insights into potential delays and providing the foundation for more effective airport management strategies to enhance on-time performance.

In (Mamdouh et al., Mar. 2024), the authors addressed the increasing importance of accurate aircraft delay forecasting in the aviation industry. They emphasised that multi-billion-dollar losses faced by airline airports and the potential loss of passenger loyalty make precise flight delay prediction a critical research issue for improving overall service performance and providing dependable travel itineraries. The authors proposed a comprehensive network framework named 'Attention-based Bidirectional long short-term memory' (ATT-BI-LSTM) for flight delay prediction to tackle this challenge. This framework integrates advanced machine learning techniques to enhance the accuracy of predictions. The ATT-BI-LSTM model combines several essential components: Bidirectional LSTM (BI-LSTM): This model extracts spatial and temporal information within the flight network, considering weather-related features. Attention Mechanism: An attention mechanism is introduced to enable the model to identify significant and distinguishing features that substantially impact the categorisation of flight delays. To validate the performance of the ATT-BI-LSTM framework, the authors tested it on a wide dataset of U.S. domestic flights in two different scenarios. The first scenario aimed to predict flight delay arrivals and departures by using fundamental flight features alongside weather data. In the second scenario, the focus shifted to predicting flight arrival delays with departure delays as additional features. The simulation results demonstrated the effectiveness of the ATT-BI-LSTM model. In the first scenario, training accuracy for flight delay prediction reached 88 % for both arrival and departure delays, with testing accuracies of 83 % and 82 % for departure and arrival delays, respectively. In the second scenario, testing accuracies reached 94.30 % and 93.71 % for the two datasets. The study concluded that the ATT-BI-LSTM model outperforms other models in the existing literature, suggesting that the developed framework can significantly contribute to mitigating flight delays by providing a highly accurate real-time prediction system for airport and aviation authorities.

Also, in (K. Kiliç and J. M. Sallan, "Study of delay prediction in the US airport network," Aerospace, vol. 10, no. 4, Apr., 2023), the authors leveraged Artificial Intelligence (AI) and Machine Learning (ML) to address the issue of predicting arrival flight delays within the United States airport network. Flight delays are known to have significant social, environmental and economic impacts, making their prediction and mitigation crucial in the aviation industry. The authors use ML models to reduce the adverse effects of these delays on the operational decision-making process. To achieve this goal, the authors conducted a comprehensive literature review and critical appraisal of previous research on flight delay prediction. They analysed various aspects of previous studies, such as the datasets used, selected features, algorithms employed and evaluation tools utilised. This analysis informed the methodology adopted in their research. The authors collected data from two publicly available datasets containing domestic flight and weather information from 2017 for their study. The data were meticulously processed and divided into training, validation and testing sets to facilitate model training and evaluation. Subsequently, they applied various ML models to the data and assessed their performance based on specific performance metrics using the testing data. Following their evaluation, the authors concluded that the Gradient Boosting Machine outperformed other ML models, such as logistic regression, random forest and feed-forward neural networks, in predicting flight delays. This finding suggests that the Gradient Boosting Machine is a promising approach for addressing the complex task of flight delay prediction in the United States airport network. By leveraging AI and ML in this manner, the aviation industry can enhance its ability to make informed and timely decisions to mitigate the impact of flight delays.

Our paper proposes a novel method by extending the state-of-the-art DeepONet, flight time reliability and airport network disruption literature. The contribution listed in the introduction can be emphasised as follows: First, we indicated the importance of reliability to all the aviation industry stakeholders. Second, we demonstrate how single airline disruption can negatively affect passenger satisfaction and cause airlines huge losses and flight itinerary schedules. Third, our novel model shows superior results through our extensive experiments by reducing the delay time to the barest minimal. The evaluation results using the different error measures in our models reveal how important having a small error of prediction on all the metrics and the lessons the passenger and airline industry can learn from predicting, mitigating and navigating flight delays in the future.

3. Problem definition

According to the Federal Aviation Administration (FAA), any flight that departs or arrives 15 min later than the scheduled time is considered late or delayed. Several factors cause this delay at the origin or destination airports. The FAA has categorised the delays as NAS, carrier, weather, security and late aircraft. Assume T is the scheduled time for departure or arrival of flight F. Flight F is considered delayed if it takes off or arrives at the Origin-Destination airport at a time $T > 15$ min. A delay is considered severe based on how later than 15 min the flight arrives or departs; a delay is small if it is as small as $T = 5$ min to $T = 20$ min or severe if it is up to $T = 65$ min. All these factors play a significant role in airport departure or arrival delays; a flight may encounter a mild or severe delay under any given conditions. Therefore, this research aims to develop a novel deep-learning approach using DeepONet to predict flight delay for a month.

3.1. Proposed GMOA-based DeepONet for prediction of flight delay

The rationale for choosing DeepONet is due to its ability to model complex relationships and handle high-dimensional data. The integration of the Gradient Mayfly Optimisation Algorithm facilitate the process of training and fine-tuning the DeepONet to ensure it performs well on our task with efficient optimisation strategy, allowing for improved convergence, minimal error and reduced training time, critical in our research, where accurate and timely predictions are vital. Also, DBN is included as a benchmark due to its established success in modelling hierarchical features, making it relevant for comparison, especially in understanding how our proposed approach compares feature extraction

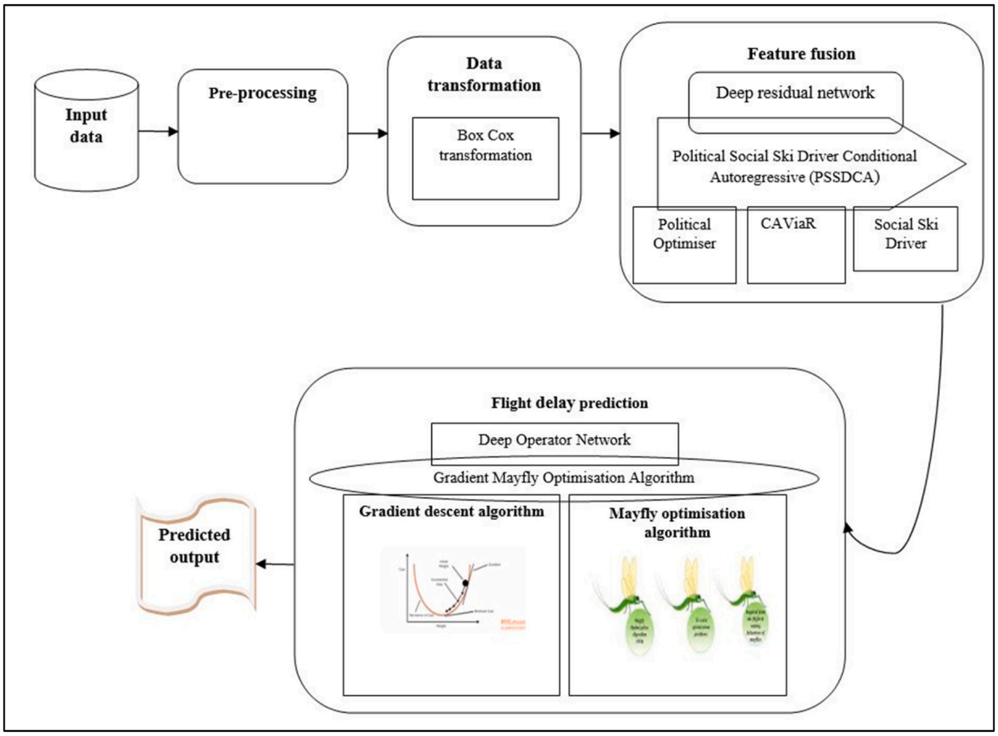


Fig. 1. Structure of flight delay prediction model using proposed GMOA-based DeepONet.

Table 1
Dataset description.

S/ No.	Feature	Data type	Description
1	YEAR	int64	Year of the flight
2	MONTH	int64	Month of the flight
3	DAY_OF_MONTH	int64	Day of the month of the flight
4	FL_DATE	object	Flight date
5	OP_CARRIER AIRLINE_ID	int64	Operating carrier's airline ID
6	OP_CARRIER_FL_NUM	int64	Operating carrier's flight number
7	ORIGIN_AIRPORT_ID	int64	Origin airport's ID
8	ORIGIN_STATE_ABR	object	Abbreviation of origin state
9	DEST_AIRPORT_ID	int64	Destination airport's ID
10	DEST_CITY_MARKET_ID	float64	Destination city market ID
11	DEST_STATE_ABR	object	Abbreviation of destination state
12	CRS_DEP_TIME	float64	Scheduled departure time
13	DEP_TIME	int64	Actual departure time
14	DEP_DELAY	float64	Departure delay in minutes
15	WHEELS_OFF	float64	Actual wheels-off time
16	WHEELS_ON	float64	Actual wheels-on time
17	CRS_ARR_TIME	Int64	Scheduled arrival time
18	ARR_TIME	float64	Actual arrival time
19	ARR_DELAY	float64	Arrival delay in minutes
20	CARRIER_DELAY	float64	Delay caused by the carrier

and representation. Gradient boosting is a popular ensemble method known for its predictive power and robustness. Including this as a benchmark allows us to evaluate whether our proposed GMOA-based DeepONet approach outperforms or is comparable to this widely-used technique. Information Gain-SVM is included for comparison as it is a traditional, well-established classification method. Comparing our proposed approach to this conventional technique helps assess the advancements achieved through GMOA-based DeepONet. The multi-agent approach is considered due to its relevance in complex system modelling. Its inclusion enables us to evaluate how our proposed approach, based on DeepONet with GMOA, fares against a model designed for multi-agent systems. LSTM is widely recognised for its efficiency in modelling sequential data. Comparing our GMOA-based DeepONet approach with Deep LSTM allows us to assess its effectiveness in capturing temporal dependencies and predicting flight delays. The SSDCA-based Deep LSTM is included to evaluate our approach against a model incorporating social dynamics. This allows us to understand if our proposed GMOA-based DeepONet approach can outperform or complement a model considering social influence. This benchmark method is crucial for a direct comparison with our proposed GMOA-based DeepONet approach, allowing us to evaluate the impact of GMOA on DeepONet's performance and its potential advancement. By including these diverse benchmark methods, we comprehensively assess our proposed GMOA-based DeepONet approach and demonstrate its comparative advantages in accuracy, efficiency, and applicability in predicting flight delays. This comparative analysis help to validate our novel approach effectiveness and identify its potential for practical deployment in the aviation domain.

The delay in flights hurts passengers and airlines. The forecasting of delays is essential in aviation. In addition, the design of precise prediction techniques for predicting flight delays becomes complex because of complications of air transportation models and the accumulation of flight data. The goal is to devise a technique for predicting flight delay considering GMOA-based DeepONet. Initially, the input data is subjected to data pre-processing to discard the data redundancies from the data. After that, the data transformation is done using Box-Cox transformation to make it suitable for improved processing. The feature fusion is done with DRN, wherein the training of DRN is done with PSSDCA. The PSSDCA is obtained by integrating SSD, CAViaR and PO. The flight delay prediction is performed with DeepONet, trained with GMOA. The proposed GMOA combines Stochastic Gradient Descent (SGD) and the mayfly algorithm (MA). Fig. 1 shows the structure of the flight delay prediction model using the proposed GMOA-based DeepONet.

Consider a database M in which the input data is expressed as Z considering various attributes, which is given by Equation (1):

$$M = \{Z_{f,g}\}; (1 \leq f \leq F); (1 \leq g \leq G) \tag{1}$$

Table 2
Selected features for training.

S/ No.	Feature	Data type	description
1	MONTH	int64	Month of the flight
2	DAY_OF_MONTH	int64	Day of the month of the flight
3	OP_CARRIER AIRLINE_ID	int64	Operating carrier's airline ID
4	OP_CARRIER_FL_NUM	int64	Operating carrier's flight number
5	ORIGIN_AIRPORT_ID	int64	Origin airport's ID
6	DEST_AIRPORT_ID	int64	Destination airport's ID
7	CRS_DEP_TIME	float64	Scheduled departure time
8	DEP_TIME	int64	Actual departure time
9	DEP_DELAY	float64	Departure delay in minutes
10	CRS_ARR_TIME	Int64	Scheduled arrival time
11	ARR_DELAY	float64	Arrival delay in minutes
12	CARRIER_DELAY	float64	Delay caused by the carrier
13	SECURITY_DELAY	float64	Delay caused by security in minutes
14	LATE AIRCRAFT DELAY float64	float64	Delay caused by a late aircraft in minutes
15	Status	Int64	Flight status indicator (e.g., 1 for on-time, 0 for delayed)

where, $Z_{f,g}$ signifies data present in dataset M signifying g^{th} attribute of f^{th} data and F specify total data, and G signifies total attributes.

3.2. Dataset description

We downloaded the necessary data for our analysis from the United States of America Bureau of Transportation Statistics (BTS) website (United States Department of Transportation, "Bureau of Transportation Statistics, Bureau of Transportation Statistics," United States Department of Transportation. Accessed: Mar. 03, 2020). The dataset employed contains domestic flight on-time information for January 2021 and November 2021, respectively, in a CSV format. In the January 2021 data, the row size is 361428, and the column size is 20. In November 2021 data, the row size is 547559. The data contained different attributes related to flight delays. These attributes included departure time, departure delay, carrier delay, airline ID, airport IDs and other data. The data was downloaded and stored locally for further analysis. After reading all the CSV files, the separate DataFrames were merged into a single consolidated DataFrame using the concat function from the Pandas library. It resulted in a comprehensive dataset containing January and November 2021 flight information. Table 1 shows the detailed descriptions of the dataset attributes. The target attributes are considered delays.

3.3. Pre-processing for noise removal

Pre-processing is a necessary step to inspect the data. The significance of pre-processing is to offer smooth and suitable for enhanced processing of input data. Before proceeding with the analysis, the collected flight delay data needed pre-processing to ensure the data was suitable for model training and prediction. Initially, columns such as'YEAR' and other irrelevant ones were removed using the drop() function from the Pandas library. This step aimed to eliminate redundant information that would not significantly contribute to the prediction process. We then dropped columns deemed irrelevant for the predictive task were also dropped from the DataFrame, including'DIV AIRPORT LANDINGS','DIV REACHED DEST','ARR DEL15′,'DEP DEL15′,'ARR TIME BLK','DEP DEL15′,'DEP DELAY GROUP', and'DEP TIME BLK'. As shown in Table 1, from the features listed, when it comes to airport data, only'ORIGIN AIRPORT ID' and DEST AIRPORT ID were selected for training, all the other airport data were dropped as it was all similar data which provided unique IDs with different names.

After clearing the unwanted data, we replaced any NaN values in the various delay columns with zeroes to handle missing values in the dataset. The values were replaced with zeroes as where there was no delay, the data had missing values as it was possibly not registered as a delay. This ensures that missing values in these columns do not affect subsequent analyses and predictive tasks. Furthermore, any rows containing missing values in any column were dropped from the DataFrame using the dropna() function with the axis parameter set to 0. This step was taken to ensure the dataset used for prediction was complete and free of missing values.

3.4. Feature selection

We performed the feature selection process to identify the most relevant variables significantly impacting flight delay prediction. We then employed the Pearson correlation as part of the pre-processing to aid in feature selection. The Pearson correlation coefficient is a statistical technique which helps quantify the linear correlation between two variables (Montgomery et al., 2021). It is defined as the covariance ratio of two variables to the product of their standard deviations. It can be represented as shown in Equation (2):

$$R = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2}$$

R is the Pearson correlation value, n is the number of data points, x_i and y_i are the two variables of individual data points. $\bar{x}$ and $\bar{y}$ are the means of the two variables x and y, respectively. The correlation matrix is constructed based on the Pearson correlation coefficient and the interdependencies among the different features. This matrix provided valuable insights into the degree of association between each pair of features and helped identify highly correlated features that could potentially introduce multicollinearity issues or redundant information. Table 2 contains the selected features for training the model. All selected features consist of categorical and numeric variables carefully selected for training the model after using this correlation matrix. Categorical variables such as airline ID, origin airport ID and destination airport ID provided essential contextual information. Numeric variables, including departure time, departure delay, carrier delay, weather delay, security delay and late aircraft delay, were also considered influential in predicting flight delays from the correlation result. Here, the input data $Z_{f,g}$ is employed to undergo pre-processing. Thus, the output generated through pre-processed data is expressed as $(H_{f,g})_{J\times K}$ in which the matrix gives pre-processed data dimension $[J' \times K']$.

3.5. Data transformation with Box cox transformation

The pre-processed data $H_{f,g}$ Obtained from pre-processing is subjected to data transformation wherein Box-Cox transformation is adapted to perform a specific task. Data transformation is a procedure to transform data from one structure to another. Thus, the data transformation is adapted to organise data for improved processing. The transformation assists in enhancing knowledge quality. Box-Cox transformation (MacIejewski et al., 2013) transforms non-normal data into some structure of data that undergoes normal distribution. The Box-Cox transformation is applied to pre-processed data $H_{f,g}$ and is expressed by Equation (3):

$$Q = \begin{cases} \dfrac{(H_{f,g})^{\alpha} - 1}{\alpha}; \alpha \neq 0 \\ ln(H_{f,g}); \alpha = 0 \end{cases} \tag{3}$$

where Q is the outcome obtained from data transformation, and α is the transformation parameter. Thus, the output of data transformation is denoted as Q whose dimension is given by the matrix $[J \times K]$.

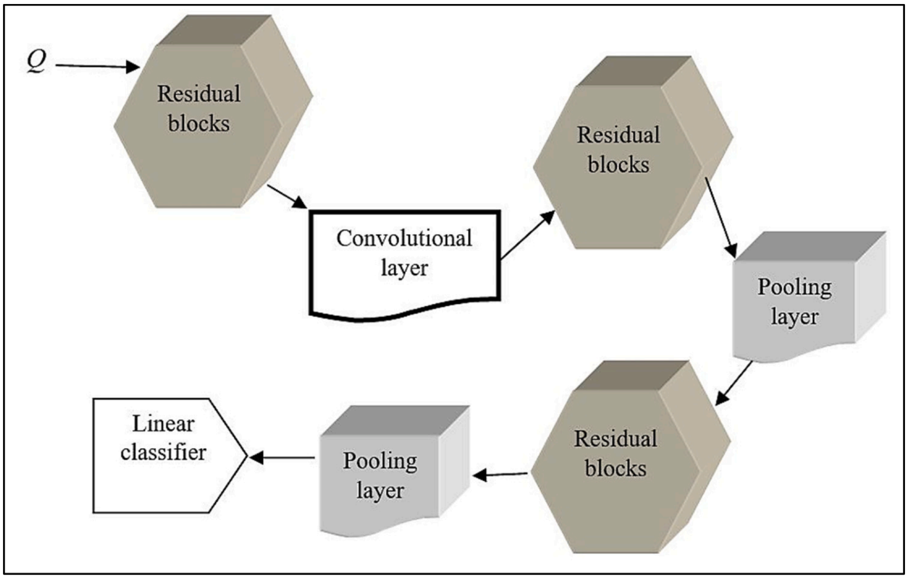


Fig. 2. Architectural model of DRN.

3.6. Fusion of features with developed PSSDCA-based DRN

After transforming data, the fusion of features is carried out with DRN (Chen et al., 2019), which is trained by developed PSSDCA using the transformed data Q. The PSSDCA is the incorporation of PO (Askari et al., 2020), SSD (Tharwat and Gabel, 2020) and CAViaR (Engle and Manganelli, 2004). After performing the transformation of data, the data dimension is $J \times K$, and thus, for minimising the features, the feature fusion step is essential to discover flight delay effectually. For instance, the G attributes have n columns; for example, n features are there and is given as $\{d_1, d_2, d_3 \cdots, d_n\}$.

3.6.1. Sorting of features with correlation

Here, the correlation is calculated for the features. For instance, the correlation is evaluated using d_r and target values. Hence, correlation values are generated. After that, the columns are altered, considering high values of correlation. Thus, to minimise the features, the fusion of features is performed.

3.6.2. Feature fusion

After the alteration of columns with correlation values, the fusion of features is done using the equation formulated as in Equation (4):

$$d_t^{fused} = \sum_{r=1}^{L} \frac{\eta}{h} d_r \tag{4}$$

where $= 1 + \frac{Q}{u}$, $h = 1tou$, Q is total features, u signifies the count of features to be chosen, and is given by Equation (5):

$$u = \frac{Q}{L}; 1 \le t \le L \tag{5}$$

where t is the fused features index and L is the total number of fused features index.

3.6.3. Discovery of η considering DRN

Once the feature fusion is done, η is discovered using DRN. The features are adapted on DRN to perform feature fusion with weights and biases based on hidden layers.

a) Structure of DRN.

Generally, the deep model offers improved accuracy and efficiency in generalising data. In addition, the deep model can deduce features automatically and can be optimally tuned to attain enhanced efficiency. In addition, the deep model is robust and generates results quickly. The DRN (Chen et al., 2019) is a promising deep neural network model in deep learning and is adapted to various pattern recognition processes. DRN poses advantages like elevated training speed, the easy transmission of Gradient, etc. The DRN can be employed to attain elevated efficiency in both regression and classification procedures. The DRN comprises various layers like the convolutional (Conv) layer, pooling layer, activation function, batch normalisation, and residual blocks. Fig. 2 displays the structure of DRN.

(i) Conv layer.

The Conv layer helps to extract effectual features automatically, considering inputted feature maps with a group of small amenable domains known as kernels. The 2D Conv layer (Conv2d) can reduce free attributes in the training process and improve the efficiency of the local receptive field and weight sharing. The Conv2d operates with the input using a sequence of filters termed kernels in a small domain with local connections. The cross-correlation of Conv2d is formulated as Equation (6) and Equation (7):

$$Conv2d(I_r) = \sum_{w=0}^{u-1} \sum_{z=0}^{v-1} Y_{w,z} \bullet I_{i_{(u+w),(v+z)}} \tag{6}$$

$$Conv1d(I_r) = \sum_{t=0}^{c_{in}-1} Y_z * I_u \tag{7}$$

Here, I_r refers to input and I_i and I_u are input with the sequence of filters, u and v, which are used for recording coordinates, Y is $u \times u$ kernel matrix and is known as a learnable attribute in the process of training, w and z express position index of 2-D kernel matrix, Y_z is kernel size of z^{th} input neuron and * is a cross-correlation operator without zero padding.

(ii) Pooling layer.

Here, the dimension evaluation is the same as that of Conv1d and Conv2d. The pooling layer is placed amongst two successive convolution layers, and it is commonly adapted to minimise the spatial size of feature maps and overwhelms the overfitting issue effectively. For functioning with individual depth and a slice of the feature map, average pooling is chosen and is given by Equation (8) and Equation (9):

$$w_{out} = \frac{w_{in} - t_w}{B} + 1 \tag{8}$$

$$z_{out} = \frac{z_{in} - t_h}{B} + 1 \tag{9}$$

where, w_{in} and z_{in} is the width and height of the input 2D matrix, w_{out} and z_{out} express as output. In addition, the height and width of kernel size are expressed as t_w and t_h.

Table 3
Discovery of training data.

k_I/d_Q	d_1	d_2	d_3	d_Q	Target
k_1	$C(k_1, d_1)$	$C(k_1, d_2)$	$C(k_1, d_3)$	$C(k_1, d_Q)$	$\eta_{training}$
k_2	$C(k_2, d_1)$	$C(k_2, d_2)$	$C(k_2, d_3)$	$C(k_2, d_Q)$	$\eta_{training}$
k_3	$C(k_3, d_1)$	$C(k_3, d_2)$	$C(k_3, d_3)$	$C(k_3, d_Q)$	$\eta_{training}$
k_4	$C(k_4, d_1)$	$C(k_4, d_2)$	$C(k_4, d_3)$	$C(k_4, d_Q)$	$\eta_{training}$

(iii) Activation function

The activation function is utilised for learning complicated nonlinear features with the original dataset and is adapted to reduce the non-linearity of mined features. The Rectified Linear Unit (ReLU) is selected as a nonlinear activation function, which is given below. Here, I_r signifies input feature maximises convergence rate and prevents vanishing gradient issue. It can be represented in Equation (10):

$$ReLU(I_r) = \begin{cases} 0, I_r < 0 \\ I_r I_r \geq 0 \end{cases} \tag{10}$$

(iv) Batch normalisation

The batch normalisation (Batch Norm) is devised for minimising the internal covariate shift by handling the input layers with scaling. This maximises the rate of learning and also vanishes gradient issues. The input I_r comprises several counts of mini-batch samples and μ_C and σ_C refers to the mean and variance of mini-batch samples. The attributes υ and ρ are employed as two learning attributes.

(v) Residual blocks.

$$C_{b,c}^{a+1} = \gamma_0 + \gamma_1 C_{b,c}^{a-1} + \gamma_2 C_{b,c}^{a-2} + \gamma_1 f\left(C_{b,c}^{a-1}\right) + \gamma_2 f \quad \left(C_{b,c}^{a-2}\right)[1 - eSin(l_1) - Sin(l_1)] + eSin(l_1)N_{b,c}^{a} + Sin(l_1)P_{b,c}^{a} \tag{16}$$

The residual block is formatted with a shortcut correlation between the input and output. Here, the inputs are linked to outputs if both acquire the same dimension. The dimension matching factor is given to matching input and output dimensions when the dimension is dissimilar, as shown by Equations (11) and (12), respectively.

$$A_1 = \zeta(I_r) + I_r \tag{11}$$

$$A_1 = \zeta(I_r) + \delta.I_r \tag{12}$$

where, I_r and A_1 is input and output of residual block, ζ signifies mapping relationship functions amongst inputs and outputs, and δ refers to dimension matching factor. Thus, the output obtained from DRN is denoted as E.

b) Training data.

The training data is essential for developing the model. After the discovery of η, the value of the training data is evaluated by correlating the means of data that belongs to a specific class. Table 3 expresses the discovery of training data.

The training of η is evaluated as in Equation (13):

$$\eta_{training}^{(r)} = Correlation(k_r, D_{k_r}) \tag{13}$$

where, D_{k_r} is the mean vector of data k_r that belongs to a class.

c) Training of DRN using PSSDCA.

The training of DRN to find η is done using the proposed PSSDCA wherein the proposed PSSDCA is devised by incorporating PO in SSDCA. SSDCA acquires the benefits of both SSD and CaViaR by providing an enhanced balance between exploitation and exploration. The method addressed highly nonlinear problems using complex constraints. It helped to address multi-objective optimisation issues. Meanwhile, PO (Askari et al., 2020) is motivated by the multiple phases of politics. The PO models the main stages of politics, like allocation of a constituency, switching of parties, election campaign, and election of inter-party and parliamentary affairs. It applies to real-world tools and can solve various engineering optimisation issues. It has a good speed of convergence and improved exploration ability in early iterations. Thus, incorporating PO in SSDCA improves the overall performance and attains a globally optimum solution. The steps of developing PSSDCA are:

Step 1) Initialisation:

The first step is solution initialisation, which is given by Equation (14):

$$C = \{C_1, C_2, \cdots, C_\kappa, \cdots, C_\vartheta\} \tag{14}$$

Where ϑ refers to total solution and C_κ signifies κ^{th} solution.

Step 2) Finding Error:

The best solution is chosen using error and is termed a minimisation issue. The minimal error reveals the best solution and is given by Equation (15):

$$MSE = \frac{1}{F}\left[\sum_{f=1}^{F} \varepsilon - E\right] \tag{15}$$

where ε refers to actual output, f is the initial value, E indicates estimated DRN output, and F is total data.

Step 3) Discovery of update equation of proposed PSSDCA:

To balance the exploration and exploitation phase, the SSDCA is adapted. Here, the updated equation of SSDCA is given as in Equation (16):

where, l_1 and l_2 are random numbers from 0 to 1. $N_{b,c}^{a}$ indicates optimal agents solution, and $P_{b,c}^{a}$ is the mean global solution for the whole population, $C_{b,c}^{a-1}$, $C_{b,c}^{a-2}$ represents solution at iteration $(a-1)$ and $(a-2)$, γ_0 and γ_1 are constants and $f(.)$ represent fitness function.

The PO is utilised to obtain enhanced optimisation ability and solve engineering issues. The updated equation as per PO is given as Equations (17) to (21):

$$C_{b,c}^{a+1} = C_{b,c}^{a-1} + m\left(C_{b,c}^{a} - C_{b,c}^{a-1}\right) \tag{17}$$

$$C_{b,c}^{a+1} = C_{b,c}^{a-1} + mC_{b,c}^{a} - mC_{b,c}^{a-1} \tag{18}$$

$$C_{b,c}^{a+1} = C_{b,c}^{a-1}(1 - m) + mC_{b,c}^{a} \tag{19}$$

$$C_{b,c}^{a+1} - mC_{b,c}^{a} = C_{b,c}^{a-1}(1 - m) \tag{20}$$

$$C_{b,c}^{a-1} = \frac{C_{b,c}^{a+1} - mC_{b,c}^{a}}{(1 - m)} \tag{21}$$

where, $C_{b,c}^{a+1}$ is the develop PSSDCA solution, $C_{b,c}^{a}$, is the solution at iteration a and m is a random number between 0, 1.

Substitute Equation (21) in Equation (16), and simplify yields Equations (22) to (26):

$$C_{b,c}^{a+1} = \gamma_0 + \gamma_1\left[\frac{C_{b,c}^{a+1} - mC_{b,c}^{a}}{(1 - m)}\right] + \gamma_2 C_{b,c}^{a-2} + \gamma_1 f\left(C_{b,c}^{a-1}\right) + \gamma_2 f\left(C_{b,c}^{a-2}\right)$$
$$[1 - eSin(l_1) - Sin(l_1)] + eSin(l_1)N_{b,c}^{a} + Sin(l_1)P_{b,c}^{a} \tag{22}$$

$$C_{b,c}^{a+1} = \gamma_0 + \frac{\gamma_1 C_{b,c}^{a+1}}{(1-m)} - \frac{\gamma_1 m C_{b,c}^{a}}{(1-m)} + \gamma_2 C_{b,c}^{a-2} + \gamma_1 f\left(C_{b,c}^{a-1}\right) + \gamma_2 f\left(C_{b,c}^{a-2}\right)[1 - eSin(l_1) - Sin(l_1)] + eSin(l_1)N_{b,c}^{a} + Sin(l_1)P_{b,c}^{a} \tag{23}$$

$$C_{b,c}^{a+1} - \frac{\gamma_1 C_{b,c}^{a+1}}{(1-m)} = \gamma_0 - \frac{\gamma_1 m C_{b,c}^{a}}{(1-m)} + \gamma_2 C_{b,c}^{a-2} + \gamma_1 f\left(C_{b,c}^{a-1}\right) + \gamma_2 f\left(C_{b,c}^{a-2}\right)[1 - eSin(l_1) - Sin(l_1)] + eSin(l_1)N_{b,c}^{a} + Sin(l_1)P_{b,c}^{a} \tag{24}$$

$$C_{b,c}^{a+1}\left[1 - \frac{\gamma_1}{(1-m)}\right] = \gamma_0 - \frac{\gamma_1 m C_{b,c}^{a}}{(1-m)} + \gamma_2 C_{b,c}^{a-2} + \gamma_1 f\left(C_{b,c}^{a-1}\right) + \gamma_2 f\left(C_{b,c}^{a-2}\right)[1 - eSin(l_1) - Sin(l_1)] + eSin(l_1)N_{b,c}^{a} + Sin(l_1)P_{b,c}^{a} \tag{25}$$

$$C_{b,c}^{a+1}\left[\frac{1-m-\gamma_1}{(1-m)}\right] = \gamma_0 - \frac{\gamma_1 m C_{b,c}^{a}}{(1-m)} + \gamma_2 C_{b,c}^{a-2} + \gamma_1 f\left(C_{b,c}^{a-1}\right) + \gamma_2 f\left(C_{b,c}^{a-2}\right)[1 - eSin(l_1) - Sin(l_1)] + eSin(l_1)N_{b,c}^{a} + Sin(l_1)P_{b,c}^{a} \tag{26}$$

The final update equation of the developed PSSDCA is given as in Equation (26):

$$C_{b,c}^{a+1} = \frac{(1-m)}{1-m-\gamma_1}\left[\gamma_0 - \frac{\gamma_1 m C_{b,c}^{a}}{(1-m)} + \gamma_2 C_{b,c}^{a-2} + \gamma_1 f\left(C_{b,c}^{a-1}\right) + \gamma_2 f\left(C_{b,c}^{a-2}\right)[1 - eSin(l_1) - Sin(l_1)] + eSin(l_1)N_{b,c}^{a} + Sin(l_1)P_{b,c}^{a}\right] \tag{27}$$

Step 4) Re-calculate error to validate feasibility:

After obtaining the updated solution, the error is re-calculated, and the solution with the least error is considered the best solution.

Step 5) Termination:

The process is repeated until maximum iteration. The pseudo-code of the PSSDCA approach is revealed in Algorithm 1.

Algorithm 1: Pseudo code of developed PSSDCA

```
Input: Solution Initialisation C
Output: Best Solution
begin
Algorithmic parameters are initialised
while stopping criteria are not met, do
for whole solutions
Re-evaluate error using equation (15)
Sort agents with error
Discover best solution
Update the SSDCA equation based on (16)
Update PO using equation (17)
Update proposed PSSDCA using equation (27)
end for
Validate feasibility of solution using error with equation (15)
return best solution
a = a + 1
end while
best solution is attained
end
```

Hence, the output generated using PSSDCA-based DRN is R with dimension $[\tau \times \ell]$.

3.6.4. *Prediction of flight delay with developed GMOA-based DeepONet*

The flight delay happens whenever the flight departs or arrives later than scheduled, resulting in huge economic losses and inconvenience. Precise prediction of flight delays is basic to form the most effective airline business. Recently, several techniques have been adapted for predicting flight delays. Here, the proposed GMOA-based DeepONet is employed for predicting the delay in flight using the fused feature R. The training of DeepONet (Lu et al., 2021) is performed with the proposed GMOA. The proposed GMOA combines MA (Zervoudakis and Tsafarakis, 2020) and the gradient descent algorithm (Wang et al., 2021). The structure of DeepONet and training of DeepONet with proposed GMOA are described in subsections.

a) Structure of DeepONet.

DeepONet (Lu et al., 2021) represents a neural network model for learning nonlinear continuous operators, like differential integrals and stochastic Ordinary Differential Equations (ODEs). It is devised based on the universal approximation theorem, which reveals that a single-layer neural network is adequate for learning a continuous nonlinear operator. To minimise the generalised error, DeepONet utilises a branch and trunk module. The training of the branch network is done for encoding the inputted function of the nonlinear operator, whereas the trunk network encodes the area of the inputted function. These two layers are combined to produce the output. The structure of DeepONet is shown in Fig. 3.

Initially, a trunk network is employed that considers i as input and outputs $[n_1, n_2, \ldots, n_p]^T \in S^p$. Moreover, in the trunk network, there exists p branch network wherein each acquires $[u(o_1), u(o_2), \cdots, u(o_m)]^T$ as input and outputs, a scalar $b_k \in S$ for $k = 1, 2, \cdots, p$. Then, the merging of them is done and is given by Equation (28):

$$\left| T(u)(i) - \sum_{k=1}^{p} \sum_{i=1}^{n} c_i^k \sigma\left(\sum_{j=1}^{m} \zeta_{i,j}^k u(o_j) + \theta_i^{ki}\right) \sigma(w_k . i + \xi_k) \right| < \varepsilon \tag{28}$$

where $\sigma(.)$ refers to a continuous non-polynomial function, n, p, m are constants, c_i^k, $\zeta_{i,j}^k$, θ_i^{ki} and ξ_k belongs to real number S, $w_k \in S^d$, $o_1 \in K_1$ that is Banach space.

The trunk network adapts activation functions in the last layer such that $t_k = \sigma(.)$ for $k = 1, 2, \cdots, p$ and hence this trunk-branch network is considered a trunk network with each weight linked to the last layer parameterised by other branch networks despite a single variable. Adding bias helps to maximise the performance by minimising the generalisation error. The addition of bias $b_0 \in S$ in the last phase, which is given by $T(u)(i) \approx \sum_{k=1}^{p} b_k t_k + b_0$. The branch networks are merged into one single branch network that results in a vector given by $[b_1, b_2, \cdots, b_p]^T \in S_p$. Thus, the output of DeepONet is denoted as U.

b) Training of DeepONet with proposed GMOA.

The DeepONet training is performed with developed GMOA and is developed by integrating the Gradient descent algorithm and MA. The MA is motivated by the flight behaviour and mating process in mayflies. The rate of convergence and speed of convergence is superior. The random flight and nuptial dance help to make better trade-offs between the exploitation and exploration phase and helps escape from the local optima. Meanwhile, the Gradient descent algorithm is an effective back-propagation that helps converge faster. Moreover, it can be utilised to discover a provided function's local minimum or maximum. It reduces the error function in an improved way. The method can handle huge datasets. Thus, combining the Gradient descent algorithm and MA helps increase the overall performance rate. The steps of the proposed GMOA are described in the next subsection.

i) Initialisation

The mayfly position in the search space indicates an impending solution. The technique undergoes certain steps. At first, two sets of mayflies are arbitrarily generated, modelling a population of males and females. Here, each mayfly is arbitrarily placed in the e -dimensional vector $W = (W_1, \cdots, W_e)$ and its efficiency is computed on specific fitness function $f(W)$. The velocity $V = (V_1, \cdots, V_e)$ of a mayfly is described as an alteration of its position and the direction of flying, considering each mayfly is devised based on flying experiences. Here, the personal best position and global best position of mayfly are given as *pbest* and *gbest*.

ii) Discovery of error.

The error of each solution is computed to find the best solution and is

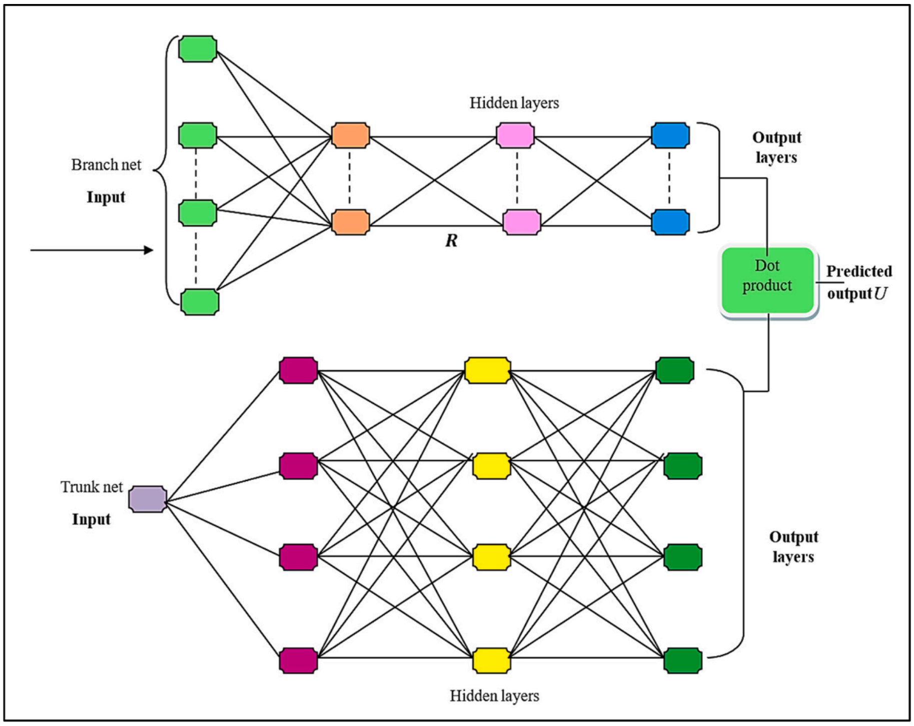


Fig. 3. Schematic view of DeepONet.

given by Equation (29):

$$MSE = \frac{1}{F}\left[\sum\nolimits_{f=1}^{F} \varepsilon - U\right] \tag{29}$$

where ε refers to estimated output and U indicates DeepONet output, and F is total data.

iii) Male mayfly's movement.

As per MA, the male mayfly position is altered using its own experience and its neighbour's experience. Consider W_a^y represents the present position of a^{th} mayfly on search space at time y and the position is altered by adding velocity V_a^{y+1} to the current position and is expressed as shown in Equation (30):

$$W_a^{y+1} = W_a^y + V_a^{y+1} \tag{30}$$

The velocity of a male mayfly is given as shown in Equation 31:

$$V_a^{y+1} = V_a^y + j_1 e^{-\beta r_p^2}\left(pbest_a - W_a^y\right) + j_2 e^{-\beta r_g^2}\left(gbest_a - W_a^y\right) \tag{31}$$

where, V_a^y signifies velocity of a^{th} mayfly at time y, W_a^y is the current position of mayfly, j_1, j_2 signifies positive attraction constants, $pbest_a$ is the best personal solution of a^{th} mayfly, and $gbest_a$ is the best global solution of a^{th} mayfly.

Here, the personal best solution is given by Equation (32):

$$pbest_a = \begin{cases} W_a^{y+1}; Iff\left(W_a^{y+1}\right) < f(pbest_a) \\ \quad keptsameOtherwise; \end{cases} \tag{32}$$

The global best solution is given by Equation (33):

$$\begin{aligned} gbest \in & \{pbest_1, pbest_2, \cdots, pbest_N | f(cbest)\} \\ & = min\{f(pbest_1), f(pbest_2), \cdots, f(pbest_N)\} \end{aligned} \tag{33}$$

where, N signifies total mayflies, β is a fixed visibility coefficient used in Equation (30), used to limit a mayfly's visibility to others, while r_p^2 is the Cartesian distance between $pbest_a$ and W_a^y and r_g^2 is the Cartesian distance between $gbest_a$ and W_a^y). These distances are calculated given by Equation (34):

$$||w_a - W_a|| = \sqrt{\sum_{v=1}^{n}\left(w_{a,v} - W_{a,v}\right)^2} \tag{34}$$

where, $w_{a,v}$ indicates v^{th} element of mayfly a and W_a is linked to $pbest_a$ or $gbest$.

The best mayflies keep altering its velocity, which is given as shown by Equation (35):

$$V_{a,v}^{y+1} = V_{a,v}^y + e^*q \tag{35}$$

where e is the nuptial dance coefficient and q is a random number between $[-1, 1]$.

Equation (30) can be given as Equation (36) to (38):

$$W_a^{y+1} = W_a^y + V_a^y + j_1 e^{-\beta r_p^2}\left(pbest_a - W_a^y\right) + j_2 e^{-\beta r_g^2}\left(gbest_a - W_a^y\right) \tag{36}$$

$$W_a^{y+1} = W_a^y + V_a^y + j_1 e^{-\beta r_p^2} pbest_a - j_1 e^{-\beta r_p^2} W_a^y + j_2 e^{-\beta r_g^2} gbest_a - j_2 e^{-\beta r_g^2} W_a^y \tag{37}$$

$$W_a^{y+1} = W_a^y\left[1 - j_1 e^{-\beta r_p^2} - j_2 e^{-\beta r_g^2}\right] + V_a^y \quad + j_1 e^{-\beta r_p^2} pbest_a + j_2 e^{-\beta r_g^2} gbest_a \tag{38}$$

The gradient descent algorithm helps to attain the optimal values quickly. As per the gradient descent algorithm, the equation is given in Equation 39:

$$W_a^{y+1} = W_a^y - \vartheta \Delta f\left(W_a^y\right) \tag{39}$$

where ϑ is a parameter that scales the Gradient as shown by Equation (40):

$$W_a^y = \vartheta \Delta f\left(W_a^y\right) + W_a^{y+1} \tag{40}$$

Substitute Equation 40 in Equation 38, yields Equations 41 to 45:

$$W_a^{y+1} = \left[\vartheta\Delta f\left(W_a^y\right) + W_a^{y+1}\right]\left[1 - j_1 e^{-\beta r_p^2} - j_2 e^{-\beta r_g^2}\right] + V_a^y + j_1 e^{-\beta r_p^2} pbest_a + j_2 e^{-\beta r_g^2} gbest_a \tag{41}$$

$$W_a^{y+1} = \vartheta\Delta f\left(W_a^y\right)\left[1 - j_1 e^{-\beta r_p^2} - j_2 e^{-\beta r_g^2}\right] + W_a^{y+1}\left[1 - j_1 e^{-\beta r_p^2} - j_2 e^{-\beta r_g^2}\right] + V_a^y + j_1 e^{-\beta r_p^2} pbest_a + j_2 e^{-\beta r_g^2} gbest_a \tag{42}$$

$$W_a^{y+1} - W_a^{y+1}\left[1 - j_1 e^{-\beta r_p^2} - j_2 e^{-\beta r_g^2}\right] = \vartheta\Delta f\left(W_a^y\right)\left[1 - j_1 e^{-\beta r_p^2} - j_2 e^{-\beta r_g^2}\right] + V_a^y + j_1 e^{-\beta r_p^2} pbest_a + j_2 e^{-\beta r_g^2} gbest_a \tag{43}$$

$$W_a^{y+1}\left[1 - 1 + j_1 e^{-\beta r_p^2} + j_2 e^{-\beta r_g^2}\right] = \vartheta\Delta f\left(W_a^y\right)\left[1 - j_1 e^{-\beta r_p^2} - j_2 e^{-\beta r_g^2}\right] + V_a^y + j_1 e^{-\beta r_p^2} pbest_a + j_2 e^{-\beta r_g^2} gbest_a \tag{44}$$

$$W_a^{y+1}\left[j_1 e^{-\beta r_p^2} + j_2 e^{-\beta r_g^2}\right] = \vartheta\Delta f\left(W_a^y\right)\left[1 - j_1 e^{-\beta r_p^2} - j_2 e^{-\beta r_g^2}\right] + V_a^y + j_1 e^{-\beta r_p^2} pbest_a + j_2 e^{-\beta r_g^2} gbest_a \tag{45}$$

The final updated equation of GMOA is given by Equation 45:

$$W_a^{y+1} = \frac{1}{j_1 e^{-\beta r_p^2} + j_2 e^{-\beta r_g^2}}\left[\vartheta\Delta f\left(W_a^y\right)\left[1 - j_1 e^{-\beta r_p^2} - j_2 e^{-\beta r_g^2}\right] + V_a^y + j_1 e^{-\beta r_p^2} pbest_a + j_2 e^{-\beta r_g^2} gbest_a\right] \tag{46}$$

iv) Female mayfly's movement

The female mayflies do not accumulate in swarms. They fly in the direction of breeding. Consider n_a^y represent the present position of female mayfly a in search space at time y and position is changed by altering the velocity V_a^{y+1} to the current position and is given by Equation (47):

$$n_a^{y+1} = n_a^y + V_a^{y+1} \tag{47}$$

Based on fitness, the best female attracts the best male, and the second-best female attracts the second-best male. Thus, the velocity can be given as Equation (48):

$$V_{a,v}^{y+1} = \begin{cases} V_{a,v}^y + j_2 e^{-\beta r_{mf}^2}\left(W_{a,v}^y - S_{a,v}^y\right); Iff(S_a) > f(W_a) \\ V_{a,v}^y + fl^* q; Iff(S_a) \le f(W_a) \end{cases} \tag{48}$$

where β is the visibility coefficient, r_{mf}^2 is the Cartesian distance among male and female mayflies as calculated in Equation (33), $S_{a,v}^y$ is the position of a female mayfly in dimension v at time step y, j_2 is a positive attraction constant, fl signifies random walk, and q refers to a random value between [-1, 1].

v) Mayflies mating.

The Crossover operator indicates the process of mating between two mayflies. Considering both female and male populations, a parent mayfly is chosen so that this choice can be done arbitrarily considering fitness. The outcomes of the crossover are given as Equations (49) and (50):

$$offspring1 = O^*male + (1 - O)^*female \tag{49}$$

$$offspring\rightleftarrows 2 = O^*female + (1 - O)^*male \tag{50}$$

Where *male* signifies the male parent, *female* specifies the female parent, O is an arbitrary value. Here, the initial velocities of the offsprings are set to 0.

vi) Compute feasibility.

The error of each solution is evaluated, and the solution with the least error is selected as the best.

vii) Termination

The steps are repeated till the best solution is generated. Algorithm 2 illustrates the pseudo-code of the designed GMOA.

Algorithm 2: Pseudo code of designed GMOA

1: Input: Population mayfly W

2: Output: W^*
3: Initialise male and female populations of mayflies
4: Evaluate error using equation (29)
5: Discover *gbest* using equation (33)
6: **do**
7: **while** stopping criteria is not satisfied
8: Update solution of male and female mayflies using equations (46) and (47)
9: Mate mayflies
10: Compute offspring using equations (49) and (50)
11: Compute feasibility using error with equation (29)
12: **end while**
13: **end do**

4. Results and discussion

We perform our experiment on PyCharm 2023.2.3 using Python v3.6.0 programming language on a Personal Computer (PC) with an Intel (R) Core (TM) i7-9700 CPU with a processor speed of 3.00 GHz and 32 GHz RAM. We used libraries; TensorFlow Core-2.4.1, TensorFlow GPU-2.4.1, NumPy-1.19.1, pandas-0.25.3, sci-kit learn-0.23.2, Scipy1.5.2, PySimpleGUI-4.29.0 and Matplolib-3.3.1.

4.1. Limitation of the study

During our research process, we were confronted with several significant challenges, including data limitations, such as data availability, quality and completeness. We initially planned to use data from the United Kingdom and the US, but we ended up using only the US data because the UK dataset was not made available, which subjected us to validate our models only on the US dataset; the complexity of our research aim and the need for well-defined methodology. These necessitate us to conduct thorough planning, data collection and a transparent acknowledgement of limitations that ensure the credibility and reliability of our research outcomes.

4.2. Performance evaluation measure

We evaluate the performance of the proposed model and other methods using the specific metrics Root Mean Square Error (RMSE), Mean Square Error (MSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). The metrics are as follows:

a) RMSE.

It is the error measure of the square root of the average square difference between actual and predicted and is given by Equation (51),

$$RMSE = \sqrt{\frac{\sum_{v=1}^{Q}\left(\ell_v - \widehat{\ell}_v\right)^2}{Q}} \tag{51}$$

Where v is variable, Q refers to the count of non-missing data, ℓ_v indicates actual data and $\widehat{\ell}_v$ refers to estimated data.

b) MSE.

It is defined as the average square measure of the target and estimated class difference, as shown in Equation (29).

c) MAE.

Mean Absolute Error is the average of the errors between all the measured values and their true values. The MAE of a prediction can be computed as in Equation (52),

$$\text{MAE} = \frac{1}{s}\sum_{j=1}^{s}\left|x_j - x\right| \tag{52}$$

Table 4
Hyperparameter settings for the methods.

S/No.	Method	Parameter settings
1	DBN	var smoothing ∈ {1e − 3, 1e − 5, 1e − 7, 1e − 9, 1e − 11}
2	Gradient boosting classifier	n − estimators ∈ {50, 100, 150, 200}, max depth ∈ {2, 5, 15, 30}, learning rate ∈ {0.001, 0.01, 0.1, 0.3}
3	Information Gain-SVM	C∈ {0.05, 0.5, 1, 5, 20, 50, 100, 300, 500}, γ ∈ {1 /n features, 1 /(n features * x.var())}
4	Multi-Agent approach	n − estimators ∈ {50, 100, 150, 200}
5	Deep LSTM	learning rate ∈ {0.001, 0.01, 0.1, 0.3}
6	SSDCA-based Deep LSTM	learning rate ∈ {0.001, 0.01, 0.1, 0.3}
7	DeepONet	batch sizes = 150, number of hidden layers = 3, learning rate ∈ {0.001, 0.01, 0.1, 0.3}, activation = relu,initializer = "Glorot normal"
8	Proposed GMOA-based DeepONet	batch sizes = 150, number of hidden layers = 3, learning rate optimal value return by GMOA

Table 5
Models cross-validation performance.

S/No.	Method	Mean MSE	Standard Error
1	DBN	0.0924	0.0102
2	Gradient boosting classifier	0.0503	0.0122
3	Information Gain-SVM	0.0344	0.0143
4	Multi-Agent approach	0.0133	0.0123
5	Deep LSTM	0.0102	0.0231
6	SSDCA-based Deep LSTM	0.0092	0.0123
7	DeepONet	0.0082	0.0273
8	Proposed GMOA-based DeepONet	0.0078	0.0101

Table 6
Special name for methods.

S/No.	Method	Special name
1	DBN	Md1
2	Gradient boosting classifier	Md2
3	Information Gain-SVM	Md3
4	Multi-Agent approach	Md4
5	Deep LSTM	Md5
6	SSDCA-based Deep LSTM	Md6
7	DeepONet	Md7

Where s denotes the total number of samples, $|x_j - x|$ is the absolute error in the prediction.

d) MAPE.

Mean Absolute Percent Error is the average of the Absolute Percent Error of the difference between the actual class and predicted class divided by the actual class. The MAPE metric can be estimated using Equation (53),

$$MAPE = \frac{1}{n}\sum_{m=1}^{n} \frac{|x_m - y_m|}{|x_m|} \times 100 \tag{53}$$

Where n indicates the total number of samples, x_m is the actual output, and y_m signifies the predicted output.

4.3. Comparison of techniques

We compared all methods with the proposed GMOA-based DeepONet approach for a comprehensive evaluation. The benchmark methods are Deep Belief Network (DBN) (Yu et al., 2019), Gradient boosting classifier (Chakrabarty, 2019), information gain-SVM (Chen et al., 2017), multi-agent approach (Guleria et al., 2019), and Deep LSTM (Kim et al., 2017), SSDCA-based Deep LSTM (Bisandu et al., 2022) and DeepONet (Lu et al., 2021).

4.3.1. Hyperparameter settings for the methods

In Table 4, we show the hyperparameters searched for the methods to select the optimal hyperparameter having a better performance. The DBN GBN with var smoothing = 1e – 7, Gradient boosting classifier with n − estimators = 50, max depth = 5, and learning rate = 0.0001, Information Gain-SVM with C = 100 and γ = 1/n features yields better performance and are selected as best hyperparameters. A hyperparameter search on Multi-Agent approach, Deep LSTM, SSDCA-based Deep LSTM, DeepONet and Proposed GMOA-based DeepONet shows that Multi-Agent approach with n − estimators = 150, Deep LSTM with first layer hidden units hidden layer1 = 35 and second layer hidden units hidden layer2 = 20, DeepOnet with hidden layers = 3, batch size of 150, learning rate = 0.001, activation = relu and the Proposed GMOA-based DeepONet with learning rate as an optimal value returned by the optimisation algorithm. The 4-fold cross-validation was applied to evaluate the model based on different training/testing sizes, as shown in Table 5.

4.4. Comparative analysis

The model's generalisation capabilities in predicting flight delays using the Deep Operator Network with Gradient-Mayfly Optimisation Algorithm are pivotal to our research. While our model exhibits remarkable performance on the available datasets, its true test lies in its ability to extrapolate to unseen or out-of-sample data, representing real-world operational scenarios. The extent of the model's generalisation to other application areas remains a subject of ongoing investigation. We acknowledge the complexities of the aviation industry, external factors and ever-changing dynamics. We emphasise that the model may require

Table 7
Results comparison of the proposed method and benchmark methods on various evaluation metric.

S/No.	Training/Testing (%)	Methods/metrics	Md1 (Yu et al., 2019)	Md2 (Chakrabarty, 2019)	Md3 (Chen et al., 2017)	Md4 (Guleria et al., 2019)	Md5 (Kim et al., 2017)	Md6 (Bisandu et al., 2022)	Md7 (Lu et al., 2021)	Proposed GMOA-based DeepONet
1	60/40	MSE	0.8223	0.5003	0.3649	0.2878	0.0387	0.0386	0.0363	0.0217
		RMSE	0.9068	0.7073	0.6041	0.5365	0.1968	0.1966	0.1905	0.1475
		MAE	0.5408	0.2805	0.1758	0.1021	0.0386	0.0218	0.0202	0.0194
		MAPE	0.2721	0.1451	0.0947	0.0557	0.0386	0.0218	0.0194	0.0186
2	70/30	MSE	0.6200	0.3783	0.2740	0.1646	0.0291	0.0283	0.0255	0.0199
		RMSE	0.7874	0.6151	0.5235	0.4057	0.1705	0.1681	0.1597	0.1413
		MAE	0.4073	0.2116	0.1329	0.0594	0.0283	0.0199	0.0152	0.0145
		MAPE	0.2046	0.1095	0.0715	0.0331	0.0283	0.0199	0.0145	0.0134
3	80/20	MSE	0.4099	0.2527	0.1836	0.1395	0.0194	0.0192	0.0189	0.0143
		RMSE	0.6403	0.5027	0.4285	0.3735	0.1392	0.1387	0.1376	0.1195
		MAE	0.2709	0.1399	0.0888	0.0492	0.0192	0.0143	0.0099	0.0096
		MAPE	0.1366	0.0723	0.0478	0.0267	0.0019	0.0149	0.0097	0.0096
4	90/10	MSE	0.2062	0.1262	0.0919	0.0385	0.0097	0.0094	0.0083	0.0058
		RMSE	0.4541	0.3552	0.3032	0.1962	0.0986	0.0969	0.0910	0.0765
		MAE	0.1354	0.0704	0.0443	0.0144	0.0094	0.0059	0.0049	0.0049
		MAPE	0.0683	0.0364	0.0238	0.0093	0.0084	0.0059	0.0049	0.0043

further refinement and real-world testing to ascertain its reliability in diverse operational environments. Nevertheless, by acknowledging these challenges and incorporating continuous learning and adaptation into the model's development, we aim to enhance its generalisation capabilities and foster a robust predictive tool for addressing the persistent challenges of flight delay management in the aviation sector.

In Table 5, we provided the 4-fold cross-validation results of the models with a comprehensive comparative analysis in terms of their Mean Squared Error (MSE) and standard error. The Deep Belief Network (DBN) exhibits the highest mean MSE of 0.0924, indicating relatively lower predictive accuracy. However, its standard error of 0.0102 implies consistent performance across folds. The Gradient Boosting Classifier follows with a lower mean MSE of 0.0503, suggesting improved predictive accuracy, but the standard error of 0.0122 signifies moderate variability. Information Gain-SVM stands out with a mean MSE of 0.0344, indicating enhanced accuracy, though with a standard error of 0.0143, signifying some unpredictability. The Multi-Agent approach excels with the lowest mean MSE of 0.0133, showcasing both accuracy and stability with a standard error of 0.0123. Deep LSTM and SSDCA-based Deep LSTM present competitive mean MSE values of 0.0102 and 0.0092, respectively, but their relatively high standard errors of 0.0231 and 0.0123 show poor reliability. DeepONet demonstrates a lower mean MSE of 0.0082, suggesting high accuracy, but an elevated standard error of 0.0273 indicates substantial variability. The Proposed GMOA-based DeepONet outperforms all models with the lowest mean MSE of 0.0078 and a standard error of 0.0101, showcasing precision and stability. In summary, the Multi-Agent approach and Proposed GMOA-based DeepONet are the relatively best models, combining accuracy and stability. DeepONet, Information Gain-SVM and the Gradient Boosting Classifier exhibit strong accuracy but need enhanced stability. DBN, Deep LSTM, and SSDCA-based Deep LSTM require improved accuracy and stability for practical use.

Our experiment evaluated the GMOA-based DeepONet model with other benchmark methods for flight delay prediction tasks using the

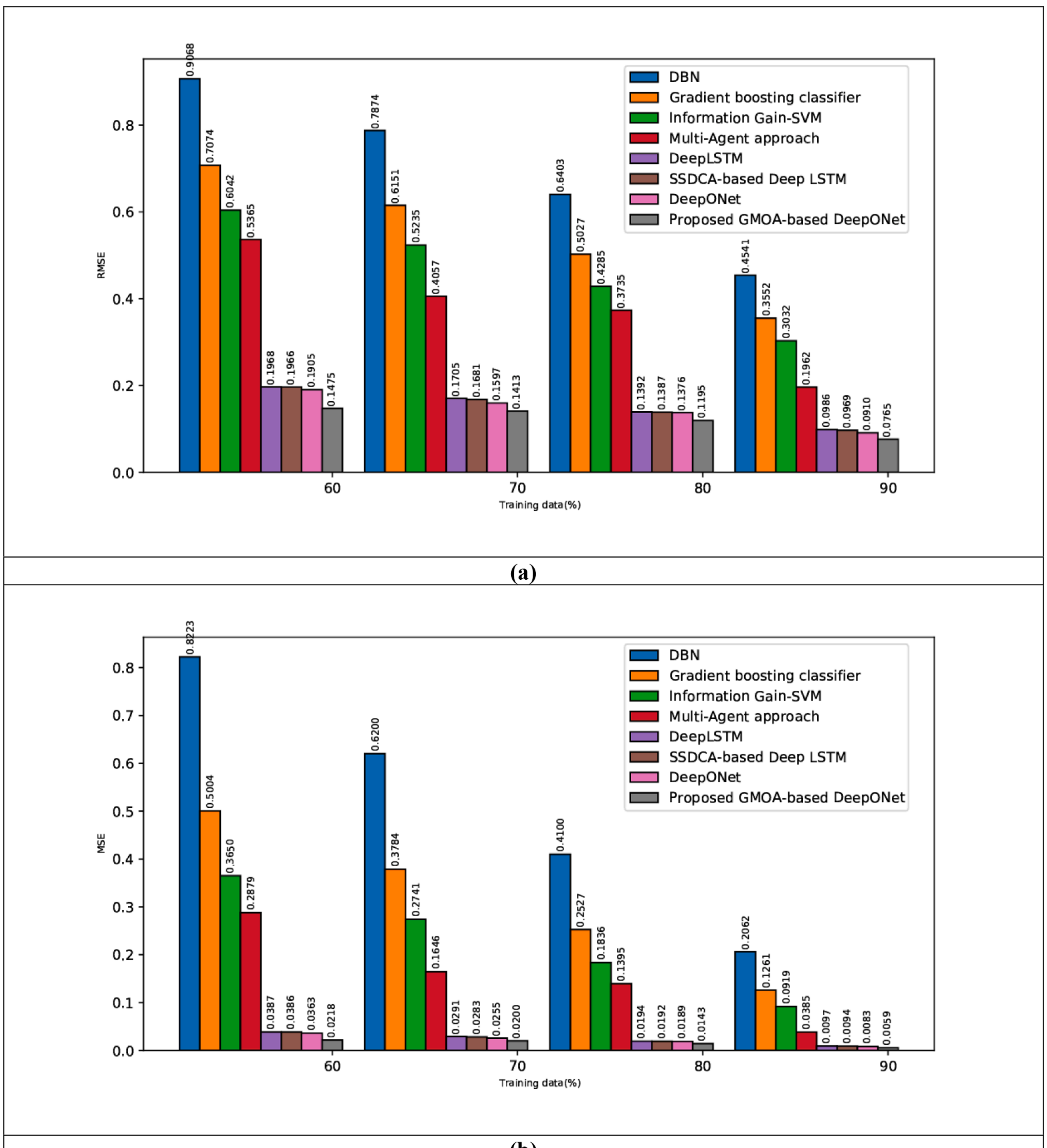
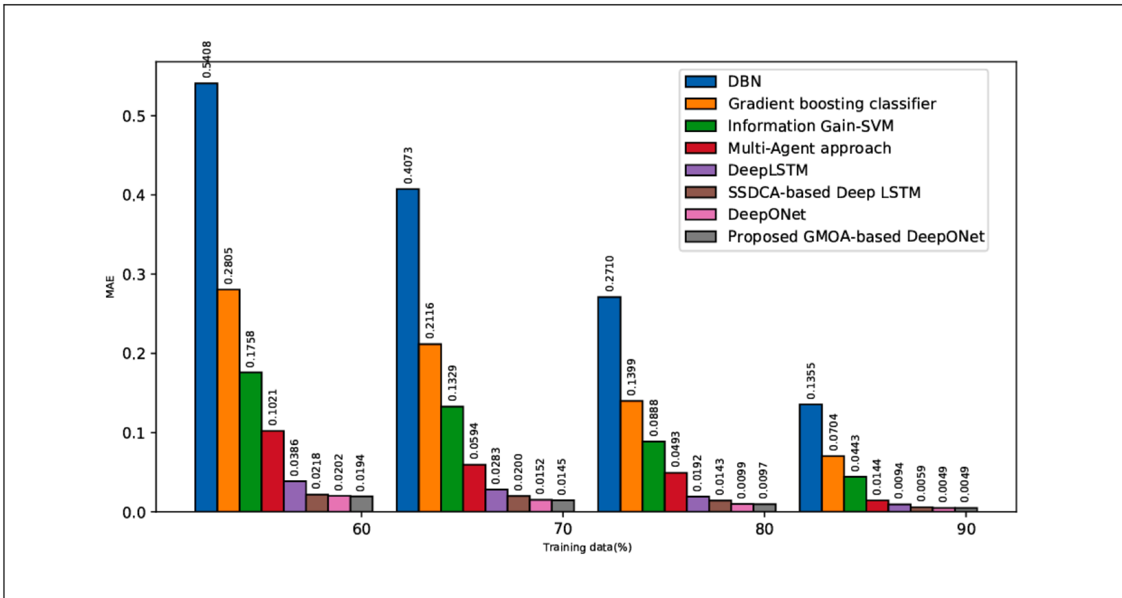
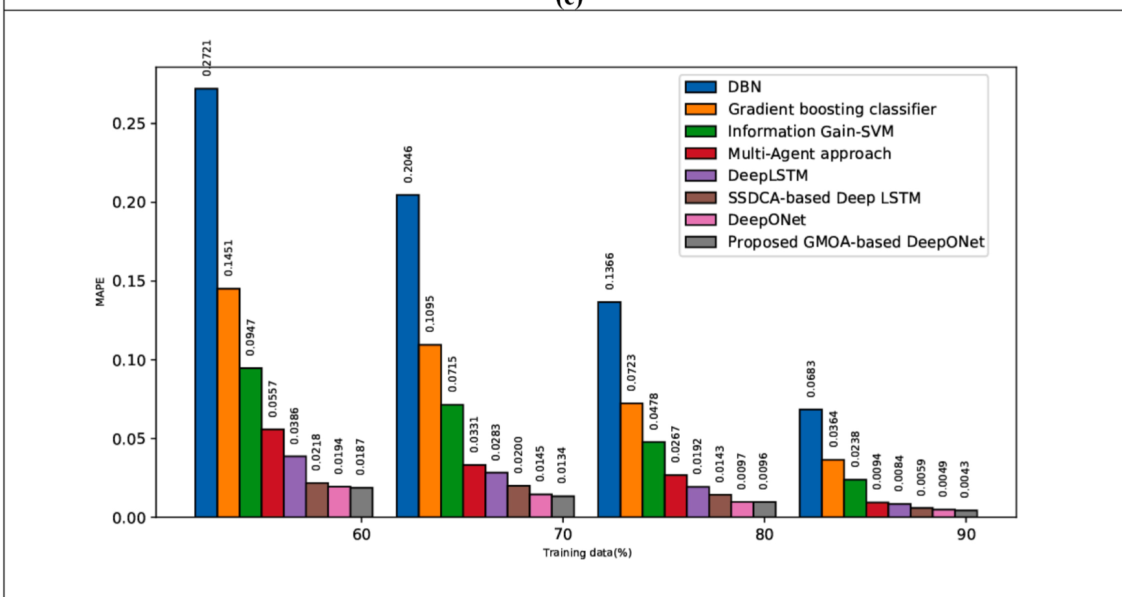


Fig. 4. Assessment of techniques using a) RMSE, b) MSE, c) MAE and d) MAPE.



(c)



(d)

Fig. 4. (*continued*).

evaluation metrics defined in Section 4.2. We perform the analysis using the January and November 2021 datasets, respectively. We further named the methods with special names for simplicity, as shown in Table 6.

4.4.1. *Assessment using January 2021 data*

Table 7 displays the experimental results from the evaluation of models on the January 2021 dataset with all the training and testing sets split ratios; the bolded black are the best performance values across all techniques. Fig. 4 (a-d) shows the performance results plots for all the techniques using the specific evaluation metrics, considering the RMSE metric for evaluation, as seen in Fig. 4a. For the training and testing set split of 60:40 per cent, the performance of all the models, the performance error for four of the benchmark methods are above 0.500, for DBN is 0.906, the Gradient boosting classifier is 0.707, Information Gain-SVM is 0.604, and the Multi-Agent approach is 0.537. While the Deep LSTM is 0.198, SSDCA-based Deep LSTM is 0.197, and DeepONet is 0.191, indicating more improvements of more than 0.200 from the other methods. However, the goal is to have a model with as minimum prediction error as possible because of the high risk associated with a problem like flight delays. Our developed GMOA-based DeepONet further improves by reducing the prediction error to 0.148 out of 1.000. Also, for a dataset split of 70:30, 80:20 and 90:10 training/testing set percentages, the RMSE of our proposed GMOA-based DeepONet consistently shows improvements better than the baseline methods with values of 0.0143, 0.119 and 0.076 as the best minimal error of predictions.

Furthermore, we needed to validate the proposed GMOA-based DeepONet on other performance evaluation metrics because of the differences in computing their errors for vast performance confirmation. To do that, we analysed the models with three other metrics mentioned in Section 4.3 (MSE, MAE and MAPE, as seen in Fig. 4b-4d). For the dataset split for the 60:40 per cent training and testing set, the proposed GMOA-based DeepONet outperformed the other methods on all evaluation

Table 8
Results comparison of the proposed method and benchmark methods on various evaluation metric.

S/ No.	Training/ Testing (%)	Methods/ metrics	Md1 (Yu et al., 2019)	Md2 (Chakrabarty, 2019)	Md3 (Chen et al., 2017)	Md4 (Guleria et al., 2019)	Md5 (Kim et al., 2017)	Md6 (Bisandu et al., 2022)	Md7 (Lu et al., 2021)	Proposed GMOA-based DeepONet
1	60/40	MSE	0.7799	0.4796	0.3810	0.3407	0.0597	0.0597	0.0591	0.0486
		RMSE	0.8831	0.6926	0.6172	0.5834	0.2443	0.2443	0.2430	0.2204
		MAE	0.5105	0.2699	0.1738	0.1350	0.0597	0.0486	0.0299	0.0298
		MAPE	0.2553	0.1424	0.0975	0.0735	0.0596	0.0486	0.0298	0.0296
2	70/30	MSE	0.2749	0.1497	0.1016	0.0492	0.0291	0.0248	0.0234	0.0209
		RMSE	0.7556	0.6016	0.5887	0.5056	0.2116	0.2116	0.2069	0.1919
		MAE	0.3825	0.2029	0.1311	0.1210	0.0447	0.0368	0.0239	0.0223
		MAPE	0.1915	0.1071	0.0734	0.0646	0.0448	0.0368	0.0223	0.0220
3	80/20	MSE	0.3884	0.2423	0.1855	0.1703	0.0299	0.0298	0.0288	0.0189
		RMSE	0.6232	0.4922	0.4307	0.4127	0.1728	0.1728	0.1697	0.1376
		MAE	0.2549	0.1345	0.0877	0.0661	0.0299	0.0189	0.0155	0.0149
		MAPE	0.1276	0.0710	0.0491	0.0362	0.0299	0.0189	0.0149	0.0147
4	90/10	MSE	0.1947	0.1205	0.0856	0.0749	0.0149	0.0149	0.01448	0.0120
		RMSE	0.4412	0.3471	0.2926	0.2737	0.1222	0.1222	0.1204	0.1097
		MAE	0.1277	0.0674	0.0437	0.0272	0.0149	0.0120	0.0079	0.0075
		MAPE	0.0640	0.0355	0.0244	0.0152	0.0149	0.0120	0.0074	0.0074

metrics, with the best performance value of 0.018 on the MAPE. It is worth noticing that some models have the same performance despite the computation methods' differences. For example, Md5 and Md6 on metric MAE and MAPE obtained values of 0.038 and 0.021. These similarities could be due to the insensitivity of MAE and MAPE to outliers because they are measures for absolute differences in error. This implies that the models learn from the training samples at the same rate as during the training. Also, for dataset splits for 70:30, 80:20 and 90:10 percentages, we observed similarity in Md5 and Md6 on MAE and MAPE metrics. The reason for the similarity may not be far from the sensitivity towards outliers.

The prediction error of the GMOA-based DeepONet reduced consistently in a better pattern than other proposed methods, with best values of 0.0134, 0.009 and 0.004 for the remaining dataset percentage splits. For this study, minimal prediction error represents the rate at which the model correctly predicts flight delays. In commercial aviation, it is important for resource allocation, improving passenger experience, key performance indication and reducing unwanted loss. The consequences of not being aware of the change in flight schedules may result in missing huge contracts by passengers, huge financial loss and poor utilisation of resources. This is much more serious than being delayed at the terminal due to security checks. Therefore, the error rate is an essential indicator in flight delay prediction and analysis. The error varies due to the different computation approaches between actual and predicted. Some take the average of the total sample, while others take only the sum of the difference between the actual and predicted. Considering the four evaluation metrics, we can say that when the datasets are divided into the four training and testing ratios, the overall model effect of the proposed GMOA-based DeepONet is better than the other models. The proposed GMOA-based DeepONet achieved a minimal error of 0.0180 on 60:40 but also continued to reduce the error as the training percentage increased on all metrics.

4.4.2. Assessment using November 2021 data

Table 8 shows our analysis of techniques on the November 2021 dataset using the metrics RMSE, MSE, MAE and MAPE, as shown in Fig. 5(a-d). We evaluated the model's effect with RMSE, as seen in Fig. 5a. When we split our training and testing set 60:40 per cent, it shows that Md1, Md2, Md3 and Md4 have a high error of more than 0.500 out of 1.000. While Md5, Md6 and Md7 have an error rate below 0.500. However, the Proposed GMOA-based DeepONet outperforms the other methods with a minimal error of 0.220 on RMSE and minimal error of values of 0.048, 0.029 and 0.029 for MSE, MAE and MAPE metrics, respectively, first, regarding the importance of the training/testing splits as observed in Fig. 5(b-d). Evaluating the models further for 70:40, 80:20, and 90:10 training/testing dataset percentage splits using all the evaluation metrics to measure the performance of the models. However, the other models' error rate shows some decrease. However, the GMOA-based DeepONet still outperforms them with values 0.029, 0.147 and 0.0074 for MAPE on the training/testing splits. It shows that the other models have less result quality from the dataset partition than the developed GMOA-based DeepONet, which consistently shows high differences in decreasing error rates. Indicating that all the predicted flights have higher reliability is significant for flight delay prediction. Therefore, we can conclude that the developed GMOA-based DeepONet model's prediction results are better than the other methods. Our experiment analysis shows that there are no fixed dataset divisions. The dataset train/test partition difference changes the prediction results for different models.

We analyse the results from our experiments for the different models using different percentages of the training/test data on the evaluation metrics to explore and exploit the methods employed on the dataset. Our results clearly show that despite similarities in some results of the proposed GMOA-based DeepONet and the DeepONet. Changing the ratio ratios of the splits reveals the importance of having enough training sets because the consistent decrease in the prediction errors of the models is an improvement worth considering when developing critical decision systems such as flight delay decision systems. It is also important to note the high performance of the proposed GMOA-based DeepONet. It shows how important the variable learning rate is from the optimisation algorithm because the consistent improvement in the results against the baseline DeepONet is a result of the variation in the learning rate of the model as suggested by the optimiser at each iteration. This is also important in generalising the results as it is automated and decided from the global solution returned by the optimiser. Though the best ratio of splits is not known for every model until it is tested from the dataset's knowledge, our experiment shows the best results achieved by 90 %:10 % for all the evaluation metrics. This is probably because the higher the knowledge from the historical records for any model to learn, the better the model validation.

4.4.3. Analysis for delay prediction

***Fig. 6(a-b) shows the analysis of all techniques for predicting delays for Jan 2021 and Nov 2021. In this research, we are interested in checking our model's efficacy to predict the total number of delays for each day for Jan 2021 and Nov 2021 compared with the ground truth. The prediction of delay with Jan 2021 data is shown in Fig. 6a). When the date is 15 Jan 2021, the number of flights delayed using the Original, proposed GMOA-based DeepONet, DeepONet, SSDCA-based Deep LSTM, Deep LSTM, Multi-Agent approach, Information Gain-SVM, Gradient boosting classifier, and DBN are 1254, 1003, 1053, 877, 1239, 877, 2925, 1999 and 2706. When the date is 31 Jan 2021, the number of flights delayed using the Original, proposed GMOA-based DeepONet, DeepONet, SSDCA-based Deep LSTM, Deep LSTM, Multi-Agent approach, Information Gain-SVM, Gradient boosting classifier, and DBN are 2032, 1625, 1808, 1507, 1757, 1422, 1684, 2565 and 3337. The delay assessment with Nov 2021 data is shown in Fig. 6b). When the date is 10 Nov 2021, the number of flights delayed using the Original, proposed GMOA-based DeepONet, DeepONet, SSDCA-based Deep LSTM, Deep LSTM, Multi-Agent approach, Information Gain-SVM, Gradient boosting classifier, and DBN are 2086, 1919, 2056, 2223, 3101, 1460, 1460, 2046 and 4105. When the date is 30 Nov 2021, the number of flights delayed using the Original, proposed GMOA-based DeepONet, DeepONet, SSDCA-based Deep LSTM, Deep LSTM, Multi-Agent approach, Information Gain-SVM, Gradient boosting classifier, and DBN are 2568, 2054, 1797, 3447, 2378, 1797, 1797, 2364 and 4563. It reveals the reliability of our proposed methods for the prediction task.

4.5. Computational analysis

In Table 9, we show the detail models average computational wall time. The computational time analysis shows that our proposed GMOA-based DeepONet model needs 300 s and outperforms all other methods.

The provided computational wall times for different models offer valuable insights into their relative performance, and several noteworthy observations can be made. First, the “DeepONet” and “proposed GMOA-based DeepONet” models exhibit the longest computational times, at 573.17 and 378.73 s, respectively. These extended times could be attributed to the complexity of solving partial differential equations, which often requires substantial computational resources. On the other hand, the “SSDCA-based Deep LSTM” model also stands out with an exceptionally high time of 497.48 s, suggesting that integrating SSDCA for LSTM optimisation may introduce significant computational overhead. In contrast, models such as “Gradient boosting classifier” and “Deep LSTM” achieve relatively lower computational times at 4.13 and 7.79 s, respectively, indicating their efficiency within their respective

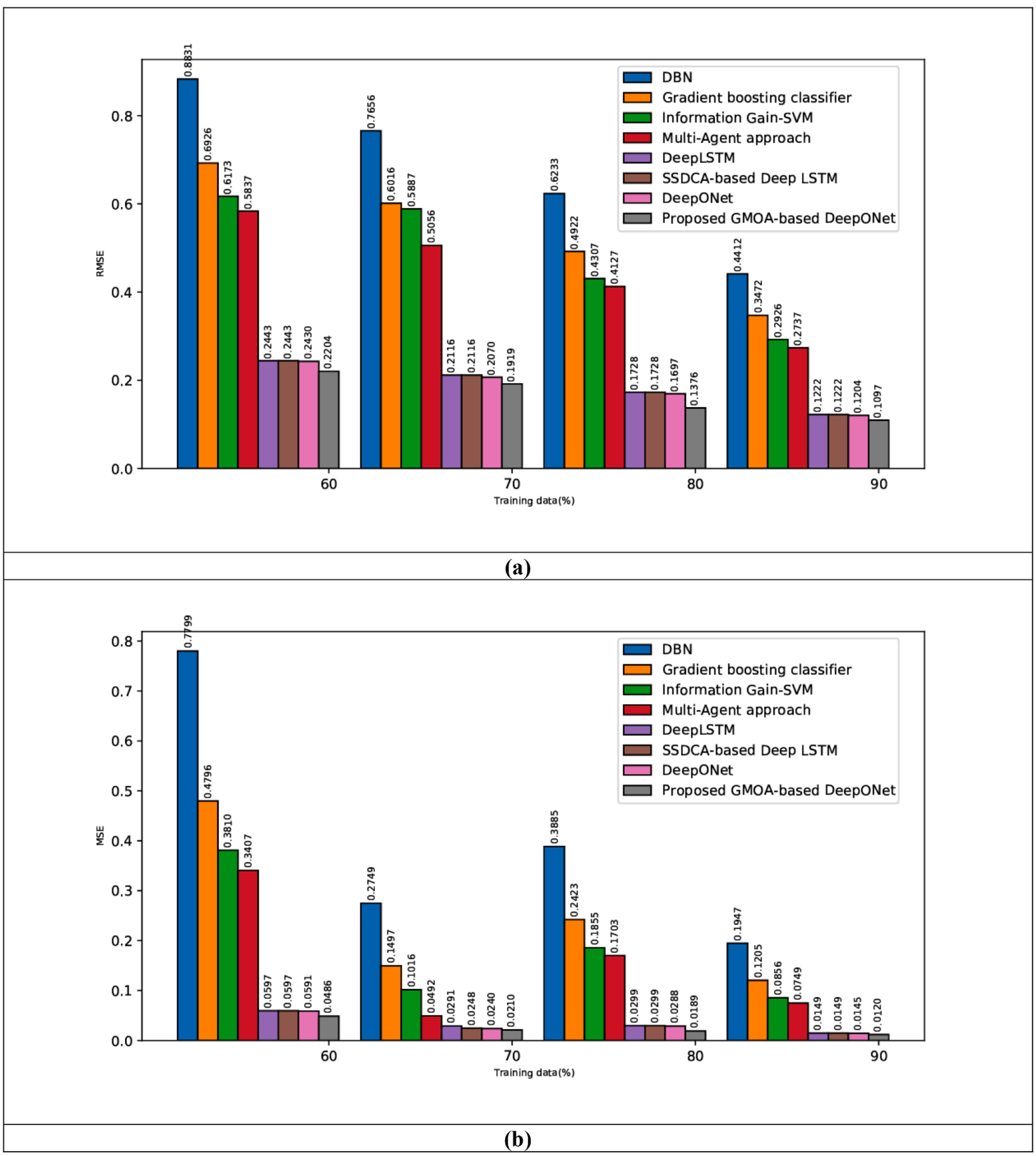
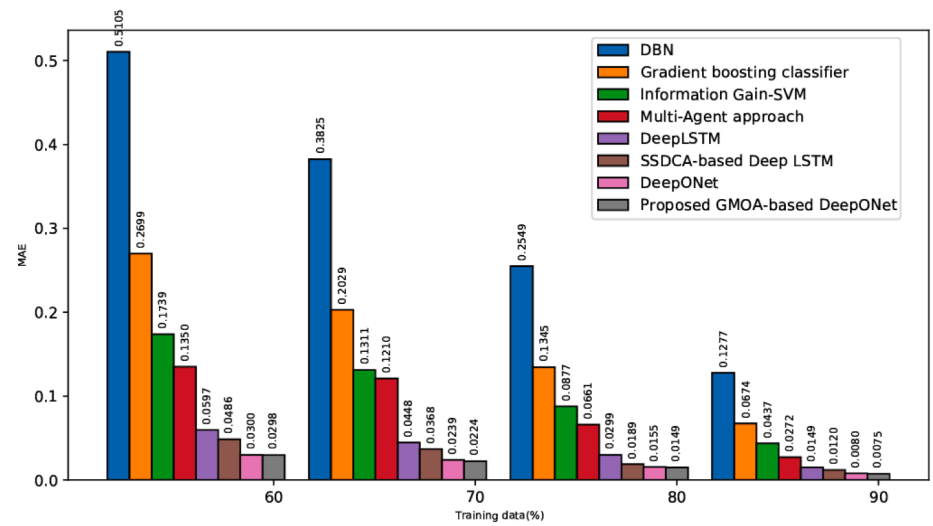
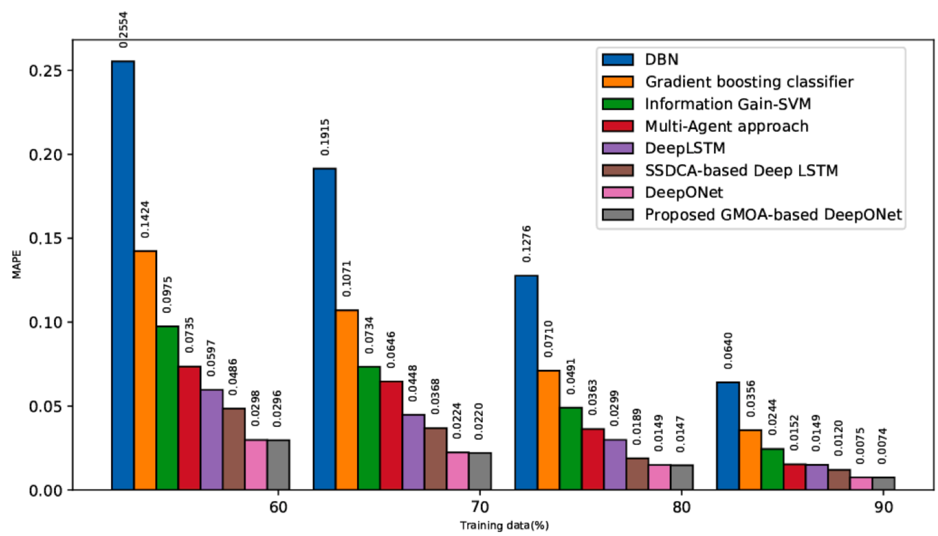


Fig. 5. Assessment of techniques using a) RMSE, b) MSE, c) MAE and d) MAPE.



(c)



(d)

Fig. 5. (*continued*).

domains. The "Information gain-SVM" and "Multi-agent approach" exhibit intermediate times, at 23.94 and 17.92 s, respectively. These times reflect the computational demands of SVM and multi-agent systems, with the "Information gain-SVM" showing a slightly higher time, likely due to its more complex nature. In summary, the choice of a model should be made in consideration of the specific application requirements. Faster computational times may be preferred when efficiency is crucial, but they should not compromise the model's ability to provide accurate and meaningful results. Conversely, models with longer computational times may be justified in applications where their inherent complexity is essential for tackling intricate problems, such as flight delay prediction. It's also worth noting that other factors, such as hardware and software infrastructure, should be considered when making model selection decisions, as they can impact the actual computational performance.

4.6. Comparative discussion of the techniques

Based on the extensive comparative analysis of Table 10, the evaluation of the proposed GMOA-based DeepONet model, as presented in the paper, demonstrates its superior performance in comparison to other benchmark methods from the literature (Chakrabarty, 2019; Zou and Hansen, 2012; Yazdi et al., 2020; Huo et al., 2020; Karádi et al., 2015; Jiang and Zhang, 2016; Zhu and Wang, 2019), The specific results provide compelling evidence of the model's performance. In the assessment conducted on the January 2021 dataset, the proposed GMOA-based DeepONet model outperforms the benchmark methods, achieving the lowest values across critical metrics: Root Mean Squared Error (RMSE), Mean Squared Error (MSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). The minimal RMSE, MSE, MAE, and MAPE values of 0.076, 0.005, 0.004 and 0.004, respectively, show the model's exceptional accuracy. These results highlight the model's efficacy in making highly precise predictions and establishing reliability. For

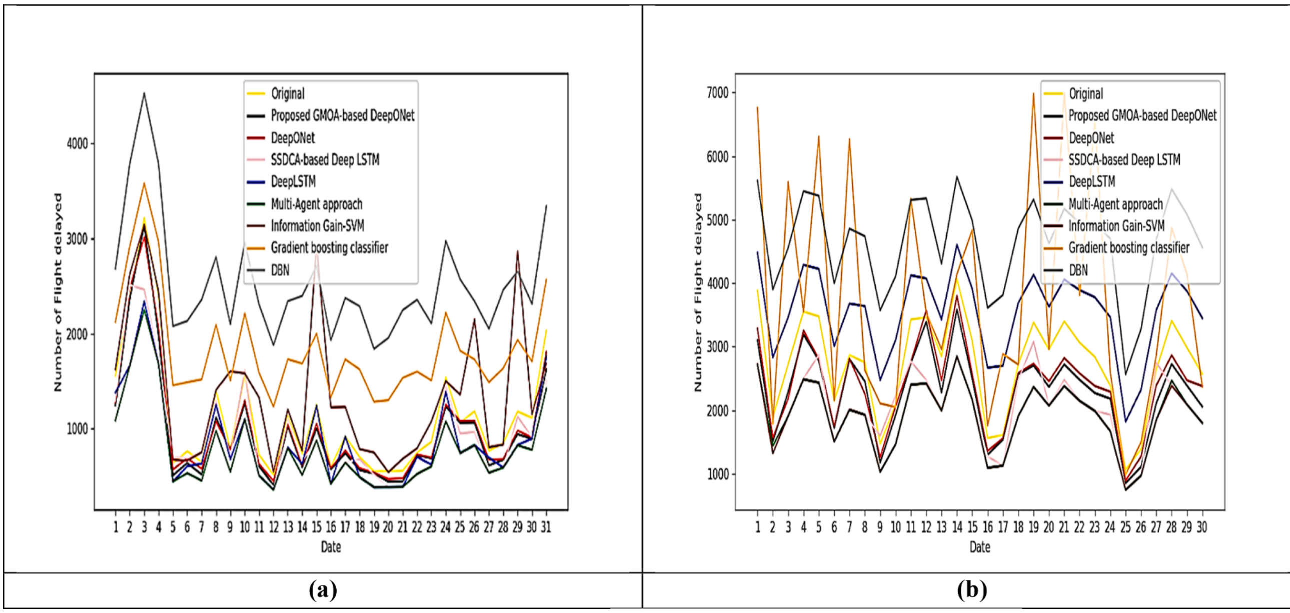


Fig. 6. Analysis of techniques using a) Jan 2021 data b) Nov 2021 data.

Table 9
Average computational wall time.

Methods	Time (Sec)
DBN	018.16
Gradient boosting classifier	004.13
Information gain-SVM	023.94
Multi-agent approach	017.92
Deep LSTM	007.79
SSDCA-based Deep LSTM	497.48
DeepONet	573.17
proposed GMOA-based DeepONet	378.73

the November 2021 dataset, the GMOA-based DeepONet model continues its impressive performance, delivering minimal RMSE, MSE, MAE, and MAPE values of 0.109, 0.012, 0.007, and 0.007. Although there is a slight increase in error values compared to the January dataset, the model maintains its superiority in the predictive task. This success is particularly notable due to the inclusion of the optimisation algorithm, which greatly enhances the model's ability to determine the global solution for learning rates. The GMOA-based DeepONet's robustness and accuracy far surpass that of the traditional DeepONet.

In the broader context of predictive accuracy, metrics such as MSE, RMSE, MAE and MAPE are important in evaluating the closeness of forecasted estimates to actual values. Lower error values indicate a closer alignment between predictions and real-world data, reflecting the model's trustworthiness. The consistently low error values the proposed GMOA-based DeepONet produced affirm its superior performance and robustness across applied datasets. It is important to acknowledge that while the GMOA-based DeepONet excels in the given datasets, benchmark methods should not be dismissed. These alternative methods may still offer promising results in different application areas. Nevertheless, the overall result from this comparative analysis strongly shows the exceptional generalisation capacity and predictive accuracy of the proposed GMOA-based DeepONet model, positioning it as a leading choice for various applications.

4.7. Practical applications, implications and benefit of findings for the aviation industry

Our research and the findings hold substantial practical applications and implications for the aviation industry. Airlines can utilise this predictive model to optimise flight schedules, reduce fuel consumption and enhance operational efficiency, leading to cost savings and improved on-time performance. Airports can better allocate resources and staff, reducing congestion, improving the passenger experience, and enhancing service quality. Passengers benefit from timely information about delays, allowing them to plan their journeys more effectively and reduce the inconvenience associated with unexpected disruptions. In the broader context, the model's implementation can minimise the economic impact of flight delays, contribute to regulatory compliance, and foster data-driven decision-making. As the model adapts to diverse operational environments, it fosters a culture of continuous improvement in the aviation industry, ultimately making air travel more efficient, reliable and passenger-centric.

Table 10
Comparative Assessment.

Data	*Metrics*	**Md1** (Yu et al., 2019)	**Md2** (Chakrabarty, 2019)	**Md3** (Chen et al., 2017)	**Md4** (Guleria et al., 2019)	**Md5** (Kim et al., 2017)	**Md6** (Bisandu et al., 2022)	**Md7** (Lu et al., 2021)	**Developed GMOA-based DeepONet**
Jan. 2021	*RMSE*	0.4541	0.3551	0.3032	0.1962	0.0986	0.0969	0.0910	0.0765
	MSE	0.2062	0.1262	0.0919	0.0385	0.0097	0.0093	0.0083	0.0059
	MAE	0.1354	0.0704	0.0443	0.0144	0.0094	0.0059	0.0049	0.0048
	MAPE	0.0683	0.0364	0.0238	0.0093	0.0084	0.0059	0.0049	0.0043
Nov. 2021	*RMSE*	0.4412	0.3472	0.2926	0.2737	0.1222	0.1222	0.1204	0.1097
	MSE	0.1947	0.1205	0.0856	0.0749	0.0159	0.0149	0.0145	0.0120
	MAE	0.1277	0.0674	0.0436	0.0272	0.0149	0.0120	0.0079	0.0074
	MAPE	0.0640	0.0356	0.0244	0.0152	0.0149	0.0120	0.0074	0.0074

5. Conclusion and lesson learned

It is critical to estimate the flight delay accurately and promptly in an intelligent aviation system. Predicting flight delays can give travellers dependable travel plans to reduce anxiety and give airports and airlines more proactive operation methods to improve service quality. This research uses flight information on multiple air route routes for flight delay prediction. We introduced a novel deep learning approach by extending a state-of-the-art method known as DeepONet for predicting flight delays by integrating the Mayfly optimisation algorithm. According to our limited knowledge, it is the first attempt at combining the DeepONet with GMOA. It is part of an ongoing research project investigating improving air passenger experience through reliable flight delay predictions based on deep learning models. We studied the efficacy of integrating DeepONet and GMOA for flight delay prediction tasks. Our results show a good fit and can achieve minimal prediction error compared with other benchmark techniques.

Moreover, we have considered only the non-weather delays at all the available airports. The Box-Cox transformation normalises the distributed data and eliminates potential collinearity among variables. The modified DRN is employed to perform the feature fusion. DRN's weight and bias update is done using PSSDCA, which is obtained by integrating SSD, CAViaR and PO. For the performance evaluation of the proposed GMOA-based DeepONet model, we utilise real-world flight data on-time from the US Transportation Bureau of Statistics in the numerical study. The new training algorithm devised for predicting the flight delay uses the DeepONet by tuning the optimal weights. Here, the weight update of DeepONet is performed with the GMOA. Compared with the other benchmark techniques, the GMOA-based DeONet model outperforms the other models. Despite the air transportation sophistication in the empirical, the GMOA-based DeepONet performs well in the flight delay prediction task with low MSE, RMSE, MAE, and MAPE. Specifically, the proposed GMOA-based DeepONet provided superior performance with the smallest RMSE of 0.085, smallest MSE of 0.007, smallest MAE of 0.005, and smallest MAPE of 0.007. The result shows that our novel GMOA proposed in this research is vital in improving flight delay prediction accuracy.

Our research contributes to the global discussions in the literature on efficient flight delay prediction models, and there are some possible areas of extension. Further openness and analysis of these trustworthy flight plans are essential for the airline industry's performance as air passengers want more information and dependability in their travel itineraries. In the future, we recommend other nature-inspired optimisation techniques and databases to check the proposed approach's reliability. Furthermore, in the data science community, the Physics-Informed-Machine-Learning (PIML) rapidly emerges, which seeks to develop a research discipline that leverages domain-specific or wealth of science to improve the effectiveness of the deep learning model. The main aim of PIML is to advance scientific understanding by generalising machine learning models and ensuring they are consistent and interpretable. Therefore, in the future, we will also attempt to use PIML methods for model development for flight delay prediction. In addition, high-performance computing, parallel computing, and cloud computing facilities widen the research scope with real-time prediction results and reduce computation time.

CRediT authorship contribution statement

Desmond Bala Bisandu: Conceptualization, Methodology, Investigation, Data curation, Writing – original draft. **Irene Moulitsas:** Conceptualization, Methodology, Investigation, Supervision, Project administration, Funding acquisition, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The source code and data used for the experiments are made freely available under the MIT license and can be downloaded from https://doi.org/10.17862/cranfield.rd.24037881.

Acknowledgement

Funding: The Petroleum Trust Development Fund (PTDF) Nigeria partially funded this work through grant number PTDF/ED/OSS/PHD/DBB/1558/19 for one of the first author's PhD studies. We also thank the UKRI for the COVID-19 recovery grant under the budget code SA077N. This research was heavily affected by the COVID-19 pandemic during the first authors' PhD studies. This lead to an extension to registration for 3 months, which was funded by the UKRI doctoral extension recovery grant.

References

Guo, Z., et al. (2020). Sgdan—A spatio-temporal graph dual-attention neural network for quantified flight delay prediction. *Sensors (Switzerland), 20*(22), 1–18. https://doi.org/10.3390/s20226433

Bubalo, B., & Daduna, J. R. (2011). Airport capacity and demand calculations by simulation-the case of Berlin-Brandenburg International Airport. *NETNOMICS: Economic Research and Electronic Networking, 12*(3), 161–181. https://doi.org/10.1007/s11066-011-9065-6

Wang, R., Shu-Li, H., Lin, Y. H., & Tseng, M.-L. (2011). "Evaluation of customer perceptions on airline service quality in uncertainty", Procedia Soc. *Behavioral Science, 25*(2011), 419–437. https://doi.org/10.1016/j.sbspro.2012.02.054

Zou, B., & Hansen, M. (2012). Flight delays, capacity investment and social welfare under air transport supply-demand equilibrium. *Transportation Research Part A: Policy and Practice, 46*(6), 965–980. https://doi.org/10.1016/j.tra.2012.02.015

Y. He, M. Hu, L. Yuan, and H. Jiang, "Flight departure time prediction based on deep learning," p. 23, 2022, doi: 10.1117/12.2623856.

J. Chen and M. Li, "Chained predictions of flight delay using machine learning," in *AIAA Scitech 2019 Forum*, 2019, pp. 1–25. doi: 10.2514/6.2019-1661.

C. Yang, Z. A. Marshall, and J. H. Mott, "A novel integration platform to reduce flight delays in the National Airspace System," *2020 Systems and Information Engineering Design Symposium, SIEDS 2020*, pp. 1–6, 2020, doi: 10.1109/SIEDS49339.2020.9106657.

Yu, B., Guo, Z., Asian, S., Wang, H., & Chen, G. (2019). Flight delay prediction for commercial air transport: A deep learning approach. *Transportation Research Part E: Logistics and Transportation Review, 125*(March), 203–221. https://doi.org/10.1016/j.tre.2019.03.013

S. Lan, "Planning for robutst airline operations: optimizing aircraft routings and flight departure times to achieve minimum passenger disruptions," 2003.

R. A. Sugara and D. Purwitasari, "Flight Delay Prediction for Mitigation of Airport Commercial Revenue Losses Using Machine Learning on Imbalanced Dataset," no. November, pp. 6–13, 2022.

Mokhtarimousavi, S., & Mehrabi, A. (2022). Flight delay causality: Machine learning technique in conjunction with random parameter statistical analysis. *International Journal of Transportation Science and Technology, 12*(1), 230–244. https://doi.org/10.1016/j.ijtst.2022.01.007

Liu, C., et al. (2023). Airport flight ground service time prediction with missing data using graph convolutional neural network imputation and bidirectional sliding mechanism. *Applied Soft Computing, 133*, Article 109941. https://doi.org/10.1016/j.asoc.2022.109941

Cao, Y., Zhu, C., Wang, Y., & Li, Q. (2019). A Method of Reducing Flight Delay by Exploring Internal Mechanism of Flight Delays. *Journal of Advanced Transportation, 2019*(5), 22–25. https://doi.org/10.1155/2019/7069380

D. B. Bisandu and I. Moulitsas, "A Deep BiLSTM Machine Learning method for flight delay prediction classification," Journal of Aviation/Aerospace Education & Research, vol. 32, no. 2, Jan. 2023, doi: 10.58940/2329-258X.1992.

D. B. Bisandu and I. Moulitsas, "A hybrid Ensemble Machine Learning approach for arrival flight delay classification prediction using voting aggregation technique," in *AIAA AVIATION 2023 Forum*, AIAA, 2023. doi: 10.2514/6.2023-4326.

D. B. Bisandu and I. Moulitsas, "A bidirectional deep LSTM machine learning method for flight delay modelling and analysis," in *The National Training Aircraft Symposium (NTAS)*, 2022, pp. 1–16. [Online]. Available: https://commons.erau.edu/ntas/2022/presentation/18/.

D. B. Bisandu, M. S. Homaid, I. Moulitsas, and S. Filippone, "A deep feedforward neural network and shallow architectures effectiveness comparison: Flight delays classification perspective," in *The 5th International Conference on Advances in Artificial Intelligence (ICAAI 2021) in QAHE at Northumbria University London Campus, UK*, ACM, 2021, pp. 1–10. doi: https://doi.org/10.1145/3505711.3505712.

Chakrabarty, N. (2019). "A data mining approach to flight arrival delay prediction for American airlines", in IEMECON 2019–9th Annual Information Technology. *Electromechanical Engineering and Microelectronics Conference*, 102–107. https://doi.org/10.1109/IEMECONX.2019.8876970
Ahmed, S. Y. (2018). Design of mobile application for travelers to transport baggage and handle Check-in process. *International Journal of Advanced Computer Science and Applications, 9*(2), 28–33. https://doi.org/10.14569/IJACSA.2018.090205
Bisandu, D. B., Moulitsas, I., & Filippone, S. (2022). Social ski driver conditional autoregressive-based deep learning classifier for flight delay prediction. *Neural Computing and Applications, 34*(11), 8777–8802. https://doi.org/10.1007/s00521-022-06898-y
Manasa, C., & Velayutham, P. (2023). Predicting flight delays with error calculation using machine learned classifiers. *Journal of Engineering Sciences, 14*(01), 93–97.
R. Khan, S. Akbar, and T. A. Zahed, "Flight Delay Prediction Based on Gradient Boosting Ensemble Techniques," pp. 1–5, 2023, doi: 10.1109/icosst57195.2022.10016828.
Faiza and K. Khalil, "Airline flight delays using artificial intelligence in COVID-19 with perspective analytics," Journal of Intelligent & Fuzzy Systems, pp. 1–23, 2023, doi: 10.3233/jifs-222827.
PeCoy, M. D., & Redmond, M. A. (2022). Flight reliability during periods of high uncertainty. *Journal of Air Transport Management, 106*(August), 2023. https://doi.org/10.1016/j.jairtraman.2022.102322
S. K. Fatima *et al.*, "A Comprehensive Framework For Model Prediction of Aerial Vehicles : Perspective Analysis and Comparison," no. January, 2023.
Chen, S., Chen, B., Shu, P., Wang, Z., & Chen, C. (2023). Real-time unmanned aerial vehicle flight path prediction using a bi-directional long short-term memory network with error compensation. *Journal of Computational Design and Engineering, 10*(1), 16–35. https://doi.org/10.1093/jcde/qwac125
Tang, Z., Huang, S., Zhu, X., Pan, W., Han, S., & Gong, T. (2023). Research on the multilayer structure of flight delay in China air traffic network. *Physica A: Statistical Mechanics and its Applications, 609*, Article 128309. https://doi.org/10.1016/j.physa.2022.128309
A. Abdelghany, V. S. Guzhva, and K. Abdelghany, "The limitation of machine-learning based models in predicting airline flight block time," *Journal of Air Transport Management*, vol. 107, no. November 2021, p. 102339, 2023, doi: 10.1016/j.jairtraman.2022.102339.
Ma, X., He, Z., Yang, P., Liao, X., & Liu, W. (2023). Agent-based modelling and simulation for life-cycle airport flight planning and scheduling. *Journal of Simulation, 00*(00), 1–14. https://doi.org/10.1080/17477778.2023.2169643
M. F. Yazdi, S. R. Kamel, S. J. M. Chabok, and M. Kheirabadi, "Flight delay prediction based on deep learning and Levenberg-Marquart algorithm," *Journal of Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00380-z.
Ding, Y. (2017). Predicting flight delay based on multiple linear regression. *IOP Conference Series: Earth and Environmental Science, 81*(1). https://doi.org/10.1088/1755-1315/81/1/012198
Ayoubi, S., et al. (2018). Machine Learning for Cognitive Network Management. *IEEE Communications Magazine, 56*(1), 158–165. https://doi.org/10.1109/MCOM.2018.1700560
Cai, K. Q., Zhang, J., Xiao, M. M., Tang, K., & Du, W. B. (2017). Simultaneous optimization of airspace congestion and flight delay in air traffic network flow management. *IEEE Transactions on Intelligent Transportation Systems, 18*(11), 3072–3082. https://doi.org/10.1109/TITS.2017.2673247
Cai, K., Jia, Y., Zhu, Y., & Xiao, M. (2015). A novel biobjective risk-based model for stochastic air traffic network flow optimization problem. *The Scientific World Journal, 2015*. https://doi.org/10.1155/2015/742541
J. Chen, H. Huang, S. Tian, and Y. Qu, "Feature selection for text classification with Naïve Bayes, Expert Systems with Applications vol. 36, no. 3 PART 1, pp. 5432–5435, 2009, doi: 10.1016/j.eswa.2008.06.054.
T. Zhang, D. Tao, X. Qu, X. Zhang, R. Lin, and W. Zhang, "The roles of initial trust and perceived risk in public's acceptance of automated vehicles," *Transportation Research Part C: Emerging Technologies* vol. 98, no. November 2018, pp. 207–220, 2019, doi: 10.1016/j.trc.2018.11.018.
Qu, J., Zhao, T., Ye, M., Li, J., & Liu, C. (2020). Flight Delay Prediction Using Deep Convolutional Neural Network Based on Fusion of Meteorological Data. *Neural Processing Letters, 52*(2), 1461–1484. https://doi.org/10.1007/s11063-020-10318-4
and M. C. Liu, Xia, Fang Wan, Lei Chen, Zhao Qiu, *Research on traffic passenger volume prediction of Sanya city based on ARIMA and grey markov models*, no. 2(153). Springer, Singapore, 2018. doi: 10.30748/soi.2018.153.08.
J. Huo, K. L. Keung, C. K. M. Lee, K. K. H. Ng, and K. C. Li, "The prediction of flight delay: Big data-driven machine learning approach," *IEEE International Conference on Industrial Engineering and Engineering Management*, vol. 2020-Decem, pp. 190–194, 2020, doi: 10.1109/IEEM45057.2020.9309919.
Zoutendijk, M., & Mitici, M. (2021). Probabilistic flight delay predictions using machine learning and applications to the flight-to-gate assignment problem. *Aerospace, 8*(6), 152. https://doi.org/10.3390/aerospace8060152
Dutrieux, S. C. M. (2021). *Predicting flight delay distributions: A machine learning-based approach at a regional airport*. the Delft University of Technology.
Ben Messaoud, M. (2021). A thorough review of aircraft landing operation from practical and theoretical standpoints at an airport which may include a single or multiple runways. *Applied Soft Computing, 98*, Article 106853. https://doi.org/10.1016/j.asoc.2020.106853
D. Karádi, C. Csiszár, and E. Nagy, "Integrated information application on mobile devices for air passengers," in *2015 International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2015*, 2015, pp. 304–311. doi: 10.1109/MTITS.2015.7223272.
Jiang, H., & Zhang, Y. (2016). An assessment of passenger experience at Melbourne Airport. *Journal of Air Transport Management, 54*, 88–92. https://doi.org/10.1016/j.jairtraman.2016.04.002
Zhu, S., & Wang, Y. (2019). Scaled sequential threshold least-squares (S 2 TLS) algorithm for sparse regression modeling and flight load prediction. *Aerospace Science and Technology, 85*, 514–528. https://doi.org/10.1016/j.ast.2018.12.038
R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning: A generic approach for extreme condition traffic forecasting," *Proceedings of the 17th SIAM International Conference on Data Mining, SDM 2017*, pp. 777–785, 2017, doi: 10.1137/1.9781611974973.87.
Pan, Z., Liang, Y., Wang, W., Yu, Y., Zheng, Y., & Zhang, J. (2019). Urban traffic prediction from spatio-temporal data using deep meta learning. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1*, 1720–1730. https://doi.org/10.1145/3292500.3330884
J. Zhang, X. Shi, J. Xie, H. Ma, I. King, D. Y. Yeung, "GaAN: Gated attention networks for learning on large and spatiotemporal graphs," *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, vol. 1, pp. 339–349, 2018.
L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators," vol. 3, no. March, pp. 31–33, 2021, doi: 10.1038/s42256-021-00302-5.
Guleria, Y., Cai, Q., Alam, S., & Li, L. (2019). "A Multi-Agent Approach for Reactionary Delay Prediction of Flights", IEEE. *Access* (vol. 7, no. Cdm,, 181565–181579. https://doi.org/10.1109/ACCESS.2019.2957874
Guvercin, M., Ferhatosmanoglu, N., & Gedik, B. (2020). Forecasting Flight Delays Using Clustered Models Based on Airport Networks. *IEEE Transactions on Intelligent Transportation Systems, 22*(5), 1–11. https://doi.org/10.1109/TITS.2020.2990960
Chen, Z., Wanke, P., Antunes, J. J. M., & Zhang, N. (2017). Chinese airline efficiency under CO2 emissions and flight delays: A stochastic network DEA model. *Energy Economics, 68*, 89–108. https://doi.org/10.1016/j.eneco.2017.09.015
Yi, J., Zhang, H., Liu, H., Zhong, G., & Li, G. (2021). Flight Delay Classification Prediction Based on Stacking Algorithm. *Journal of Advanced Transportation, 2021*. https://doi.org/10.1155/2021/4292778
Guo, Z., Yu, B., Hao, M., Wang, W., Jiang, Y., & Zong, F. (2021). A novel hybrid method for flight departure delay prediction using Random Forest Regression and Maximal Information Coefficient. *Aerospace Science and Technology, 116*, Article 106822. https://doi.org/10.1016/j.ast.2021.106822
Shao, W., Prabowo, A., Zhao, S., Koniusz, P., & Salim, F. D. (2022). Predicting flight delay with spatio-temporal trajectory convolutional network and airport situational awareness map. *Neurocomputing, 472*, 280–293. https://doi.org/10.1016/j.neucom.2021.04.136
Divya, J., Divyasindhu, P., & Joan Reshmi, L. (2023). In *Flight delay prediction using GANN - An improved Artificial Neural Network model integrating Genetic Algorithm*. Institute of Electrical and Electronics Engineers Inc.. https://doi.org/10.1109/ACCAI58221.2023.10200001
Tirtha, S. D., Bhowmik, T., & Eluru, N. (May 2023). Flight-level analysis of departure delay and arrival delay using Copula-Based Joint Framework. *Transportation Research Record, 2677*(5), 229–246. https://doi.org/10.1177/03611981221130031
Faiza and K. Khalil, "Airline flight delays using artificial intelligence in COVID-19 with perspective analytics," *Journal of Intelligent and Fuzzy Systems*, vol. 44, no. 4, pp. 6631–6653, 2023, doi: 10.3233/JIFS-222827.
Wu, Y., Yang, H., Lin, Y., & Liu, H. (2023). Spatiotemporal propagation learning for network-wide flight delay prediction. *IEEE Transactions on Knowledge and Data Engineering*. https://doi.org/10.1109/TKDE.2023.3286690
Cai, K., Li, Y., Zhu, Y., Fang, Q., Yang, Y., & Du, W. (Mar. 2023). A geographical and operational deep graph convolutional approach for flight delay prediction. *Chinese Journal of Aeronautics, 36*(3), 357–367. https://doi.org/10.1016/j.cja.2022.10.004
Khan, W. A., Chung, S. H., Eltoukhy, A. E. E., & Khurshid, F. (Jan. 2024). A novel parallel series data-driven model for IATA-coded flight delays prediction and features analysis. *Journal of Air Transport Management, 114*. https://doi.org/10.1016/j.jairtraman.2023.102488
Li, Q., Guan, X., & Liu, J. (Oct. 2023). A CNN-LSTM framework for flight delay prediction. *Expert Systems with Applications, 227*. https://doi.org/10.1016/j.eswa.2023.120287
Mamdouh, M., Ezzat, M., & Hefny, H. (Mar. 2024). Improving flight delays prediction by developing attention-based bidirectional LSTM network. *Expert Systems with Applications, 238*. https://doi.org/10.1016/j.eswa.2023.121747
K. Kiliç and J. M. Sallan, "Study of delay prediction in the US airport network," *Aerospace*, vol. 10, no. 4, Apr. 2023, doi: 10.3390/aerospace10040342.
United States Department of Transportation, "Bureau of Transportation Statistics, Bureau of Transportation Statistics," United States Department of Transportation. Accessed: Mar. 03, 2020. [Online]. Available: https://www.bts.gov/.
Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). *Introduction to linear regression analysis*. John Wiley & Sons.
MacIejewski, R., Pattath, A., Ko, S., Hafen, R., Cleveland, W. S., & Ebert, D. S. (2013). Automated box-cox transformations for improved visual encoding. *IEEE Transactions on Visualization and Computer Graphics, 19*(1), 130–140. https://doi.org/10.1109/TVCG.2012.64
Chen, Z., Chen, Y., Wu, L., Cheng, S., & Lin, P. (2019). Deep residual network based fault detection and diagnosis of photovoltaic arrays using current-voltage curves and ambient conditions. *Energy Convers Manag* (vol. 198, no. May), Article 111793. https://doi.org/10.1016/j.enconman.2019.111793
Askari, Q., Younas, I., & Saeed, M. (2020). Political Optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowl Based Syst, 195*, Article 105709. https://doi.org/10.1016/j.knosys.2020.105709

Tharwat, A., & Gabel, T. (2020). Parameters optimization of support vector machines for imbalanced data using social ski driver algorithm. *Neural Computing and Applications, 32*(11), 6925–6938. https://doi.org/10.1007/s00521-019-04159-z
Engle, R. F., & Manganelli, S. (2004). CAViaR: Conditional autoregressive value at risk by regression quantiles. *Journal of Business and Economic Statistics, 22*(4), 367–381. https://doi.org/10.1198/073500104000000370
K. Zervoudakis and S. Tsafarakis, "A mayfly optimization algorithm," *Comput Ind Eng*, vol. 145, no. September 2019, p. 106559, 2020, doi: 10.1016/j.cie.2020.106559.
S. Wang, H. Wang, and P. Perdikaris, "Improved architectures and training algorithms for deep operator networks," pp. 1–40, 2021.
Kim, S., Hong, S., Joh, M., & Song, S. K. (2017). DeepRain: ConvLSTM network for precipitation prediction using multichannel radar data. *ArXiv*, 3–6.

2024-01-31

Prediction of flight delay using deep operator network with gradient-mayfly optimisation algorithm

Bisandu, Desmond Bala

Elsevier

Bisandu DB, Moulitsas I. (2024) Prediction of flight delay using deep operator network with gradient-mayfly optimisation algorithm. Expert Systems with Applications, Volume 247, August 2024, Article number 123306

https://doi.org/10.1016/j.eswa.2024.123306