ACTA IMEKO
ISSN: 2221-870X
June 2024, Volume 13, Number 2, 1 - 15

Deep learning methodologies based on metaheuristics for predictive engine maintenance

Pradeep Kumar D[1], Sowmya B J[2], Anita Kanavalli[1], Supreeth S[3], Shruthi G[3], Rohith S[4]

[1] *Department of Computer Science and Engineering, M S Ramaiah Institute of Technology, Bengaluru, India*
[2] *Department of Artificial Intelligence and Data Science, M S Ramaiah Institute of Technology, Bengaluru, India*
[3] *School of Computer Science and Engineering, REVA University, Yelahanka, Bengaluru, India*
[4] *Department of Electronics & Communication Engineering, Nagarjuna College of Engineering & Technology, Bengaluru, 562110, India*

ABSTRACT
Recently, there has been an increase in concerns about the accessibility, security, and reliability of aviation engines. To prevent engine failures which can be quite serious, it is important to take effective measures. The objective is to create a deep learning simulation that can accurately predict an aircraft engine's viability and remaining usefulness using meta-heuristic techniques to improve its performance. These techniques discover the optimal hyper parameters and architecture for the deep learning model. This will help minimize downtime and maintenance costs for the aircraft fleet by handling complex data such as sensor readings and past maintenance records while also adapting to changing conditions over time. Since training deep learning models can be computationally intensive, meta-heuristic methods increase their robustness. The aim is to enhance performance by increasing the accuracy rate and reducing mean squared losses of multiple deep learning methods used for predicting aircraft engine maintenance by hybridizing them with metaheuristic algorithms.

Section: RESEARCH PAPER

Keywords: Predictive maintenance; deep learning; metaheuristic algorithms; remaining useful life

Citation: Pradeep Kumar D, Sowmya B J, Anita Kanavalli, Supreeth S, Shruthi G, Rohith S, Deep learning methodologies based on metaheuristics for predictive engine maintenance, Acta IMEKO, vol. 13 (2024) no. 2, pp. 1-15. DOI: 10.21014/actaimeko.v13i2.1667

Section Editor: Laura Fabbiano, Politecnico di Bari, Italy

Received September 2, 2023; **In final form** April 17, 2024; **Published** June 2024

Corresponding author: Supreeth S, e-mail: supreeth1588@gmail.com

1. INTRODUCTION

Aircraft engines are critical components of any aircraft, and their failure can result in significant safety risks, operational disruptions, and financial losses for airlines and their customers. Therefore, implementing a robust predictive maintenance programme is crucial ensuring aircraft engine reliability and safety, in the aviation sector, predictive maintenance is an important procedure, particularly when it comes to aircraft engines.

It involves the use of advanced technology, data analytics, and machine learning algorithms to predict potential issues with an engine before they occur, enabling maintenance teams to proactively address the problem before it causes unplanned downtime or a catastrophic failure. In this study, we demonstrate the design and improvement of a deep learning model that can precisely forecast the Remaining Useful Life (RUL) of aircraft engines and classify the usability of an engine, with meta-heuristic methods to optimize the model's performance. Meta-heuristic techniques find the optimal set of hyperparameters and architecture for the deep learning model, which will improve its performance on the task of predictive maintenance. Currently, fault diagnosis models are being developed to identify the root cause of failures and enable maintenance teams to address the underlying issues, as are decision support systems that can provide real-time recommendations based on sensor data and other parameters. With this model, the novelty presents itself in the combination of a metaheuristic algorithm and a deep learning framework, which, to the authors' knowledge, has not been undertaken till date for both tasks of engine viability classification and remaining useful life prediction. Secondly, this model provides a scheme for feature selection to optimize the results better while reducing the computational time. Furthermore, the model is generalized to allow for its extension to other machines whose remaining useful life is to be estimated using a host of sensor measurements like home appliances or motor vehicles.

2. RELATED WORKS

The work literature survey consists of three phases. The first phase involved examining references on the use of deep learning frameworks to aircraft engine predictive maintenance. Second phase concentrated on the use of metaheuristic algorithms in conjunction with deep learning frameworks for this task. In the third and final phase, the researchers searched for papers on the most important features to consider and challenges encountered during predictive maintenance.

The inspiration for using metaheuristic algorithms in conjunction with Deep Learning techniques for Predictive Maintenance of Aircraft engines stemmed from the research outlined in [1]. The paper proposes an improved approach for identifying engine issues in aeroplanes using a combination of the Grasshopper Optimization Algorithm (GOA) and the Echo State Network (ESN). The proposed approach involves training an ESN, a kind of recurrent neural network with a track record of success in time-series prediction challenges, on engine vibration data to predict potential faults. Then, the reservoir size, spectral radius, and feedback scaling of the GOA, a metaheuristic optimisation algorithm inspired by the swarming activity of grasshoppers, are employed to optimise the ESN's parameters. The proposed approach achieved an average prediction accuracy of 97.9 %, which is higher than the accuracy achieved by Back Propagation Neural Network (BPNN) (94.4 %), Support Vector Machines (SVM) (93.1 %), and Decision Tree (DT) (90.3 %). The ability of the GOA to identify the best parameters for the ESN is credited by the authors as the reason for the suggested approach's increased performance. In conclusion, the authors recommend the use of the proposed approach as an effective tool predicting engine faults in real-world applications.

The paper [2] showcases the development of a model comprising two deep learning techniques to accomplish the task of estimating an aircraft engine's remaining usable life, which is the number of flights an engine can take before it is deemed unfit to fly. The need for combining CNN (Convolutional Neural Network) and LSTM (Long Short-Term Memory) - CLSTM, is that the former has shown promise in extracting useful features, while the latter has showcased its strength in dealing with time series data similar to the ones used in Predictive Maintenance tasks. The authors thus conclude that the proposed CLSTM model is successful in learning the historical data well and using it to accurately predict the useful residual life of an aircraft engine. As a future scope, they have mentioned working on the design of an energy efficient-based approach.

In [3], provides an insight into understanding the trends and challenges of our problem statement. The paper covers an outline of the various machine learning, approaches used in predictive maintenance, covering more complex methods like deep learning and reinforcement learning as well as more conventional methods. The paper also discusses the challenges associated with predictive maintenance for aircraft engines, the main challenges being the significant amount of data generated by the engines, the need for a comprehensive understanding of the complex mechanisms of the engines, and lastly, those concerning data quality, data imbalance, feature selection, and the interpretability of models. The authors recommend the development of hybrid models that combine different machine learning strategies to increase the accuracy and reliability of predictive models, incorporating additional sources of data, such as maintenance records, weather data, and flight data, and lastly, developing standardized frameworks to evaluate the performance of such models, as steps to improve the current trends.

To get a feel for feature selection for this particular problem statement, [4] was consulted. In order to define the needs for Prognostics and Health Management (PHM) systems that support predictive maintenance in the aviation industry, the research study suggests a systematic methodology. PHM systems work to increase the safety and dependability of aircraft by anticipating and detecting issues, thereby enabling maintenance providers to take preventive measures. The proposed methodology consists of six phases: problem definition, stakeholder analysis, data collection, requirement identification, requirement validation, and requirement prioritization. The authors provide detailed guidelines for each phase, including the methods and tools that can be used to facilitate the process. The paper presents a case study that identifies various requirements, including the need for real-time monitoring, fault detection, and decision support tools. In conclusion, the authors recommend that the methodology proposed can be applied to other aviation systems and facilitates the development of effective PHM systems that improve the safety and reliability of aircraft operations.

In [5], provides an overview of current work on aircraft hydraulic systems and engine predictive maintenance and identifies emerging trends and challenges. By studying aircraft hydraulic data, a machine learning strategy has been applied for condition monitoring. The dataset consists of actuation system and elevator position information and is monitored to identify significant changes that may be potential indicators of ongoing failures. The problem considered in the experiment was to detect error patterns and trigger anomaly alerts when the liquid level reached critical values. The results of the window-based pattern detection for prognostics are presented in the study in order to increase accuracy and predict faults before they arise. Additionally, an LSTM-based prognostic technique for predicting aircraft faults is presented in this study. The model can predict engine behaviour with accuracy.

In the paper [6], for a fleet of aircraft, a dynamic maintenance architecture is suggested where component RUL prognostics are frequently updated. To produce RUL prognostics, a CNN is suggested. To schedule maintenance on aircraft, an integer linear programme is employed. The CNN model is applied to each of the datasets, and the obtained RUL prognostics are evaluated using the Root Mean Squared Error (RMSE) metric.

The maintenance task schedule would start as soon as the alarm is triggered. The maintenance framework parameters are determined using a genetic algorithm. Long-term statistics indicate that only 7.4 % of the overall maintenance expenditures are attributable to engine problems. The proposed maintenance planning framework can be easily applied to other aircraft components.

In [7], serves as a state-of-the-art overview for identifying new solutions being applied to predictive maintenance issues and plotting the present landscape of the field. The purpose of this overview is to identify and highlight future research encounters and prospects in this area. This is extended by identifying which predictive models and tools have been applied to these and other datasets in various PdM applications. Outlines of different projects like the Distributed Aircraft Maintenance Environment (DAME), UPTIME and in order to highlight the expansion in both academia and industry, PdM uses industrial services. The difficulties that researchers in this subject will face are imbalanced datasets, high dimensionality, industrial challenges such as start-up costs, and learning barriers. Therefore, PdM can be optimized over alternative maintenance strategies to maximize the RUL of aircraft components. These enhancements which further automate and optimise these processes, allowing aircraft operators and manufacturers to significantly cut maintenance costs.

In order to discover operational factors, a data-driven strategy is used to collect historical operational and maintenance data from an airline operator. Reliability estimates are produced using both time-independent and time-dependent Proportional Hazard Models (PHMs), strategies which include operational parameters as covariates. The proposed methodology [8] was as follows: data import for fleet-wide maintenance and flights; flight identification to determine which flights contributed to the unexpected component failure, identifying operational factors by ultimate value/maximum difference analysis, reliability modelling and future predictions. According to the results of the analysis of historical data, the frequency of unscheduled occurrences can be decreased by adopting new maintenance schedules developed from the proposed dependability models, and that they which outperform accuracy-focused time-based models.

The strategy proposed in this paper [9] is a framework for incorporating probabilistic RUL prognostics based issue was formulated while maintenance operations are initiated in response to estimations of the RUL distribution. These RUL distribution estimations enable more efficient maintenance planning. The Deep Reinforcement Learning (DRL) technique offers threshold-free, adaptive, and flexible maintenance planning as a result. Unlike the case where engines are changed at the mean-estimated-RUL, the overall maintenance cost is decreased by 29.3 % when using the DRL strategy. The engines' wasted life is also kept to just 12.81 cycles, and 95.6 % of unscheduled maintenance is avoided.

The proposed strategy [10] includes all steps necessary to carry out the RUL prediction with ambiguity and decision-making for maintenance. A LUE model (Local Uncertainty Estimation) with bidirectional long-short-term memory (Bi-LSTM) was developed to quantify the uncertainty of RUL prediction considering the prediction aspect. By linking the built-in RUL distribution to maintenance-related costs, the Maintenance Cost Rate (MCR), or maintenance cost per unit of operational time, a function is created to address the post-prediction issue. The timing pertaining to maintenance activities can be chosen by maximising the MCR function with a focus on the operational management's economic requirements. NASA's aero-engine deterioration dataset and experimental findings demonstrate the viability of the suggested data-driven technique of predictive maintenance Also, three options for the cost structure are offered to demonstrate the adaptability of the suggested technique. All sensor values are normalised using the Z-score standardisation method into a new data set with a variance of one and an average value of zero. The outcomes of 95 % confidence level online interval and point prediction for RUL.

This study [11] suggests using GRU (Gated Recurrent Unit) to identify RUL on aircraft engines in order to conduct a preventive maintenance strategy. The main takeaway from the experiment's findings is that a new method can be developed that has a faster prediction process than other methods, a more straightforward method of calculation for determining the epoch value, and results that are close to the original value in terms of both economics and RUL prediction using GRU. Long short-term memory (LSTM) neural networks, which are specialised in extracting sensor temporal information, are one of the best and most well-liked DL models to produce a prediction, especially to identify PdM on aviation engines. The training period of LSTM is substantially longer than that of other algorithms, despite the fact that its accuracy is higher than that of other algorithms. It remains difficult to figure out how to shorten training time while still guaranteeing great accuracy. One particular instance of the LSTM is the gated recurrent unit (GRU). Compared to LSTM, it requires less training time. In this paper, the GRU method uses 200 epochs and a minimal amount of compute to accurately forecast the estimated RUL of the C-MAPSS dataset.

In this research [12], an algorithm may be used to forecast the RUL. This solution has a substantially higher predictive power than earlier studies that used different methods. Only five variables had the greatest influence on the RUL, according to an examination of the variables' respective weights acting as predictors. In this study, four distinct kernel types linear, polynomial, radial basis, and sigmoidal were tested for the SVM model's training. The parameters of the hyper parameters and the selection of the ideal kernel function for each problem are crucial stages in the SVM model's training process. A substantial public database was used to test the method's validity, and the outcomes were compared to those obtained using a vector auto-regressive moving average (VARMA) model. The results reveal that the suggested model outperformed the VARMA model by a wide margin.

The paper proposes a framework [13] for foreseeing incredibly uncommon failure events in aircraft maintenance using DRL. The proposed framework uses historical maintenance data to identify critical variables that affect the occurrence of rare failures and to provide recommendations for resource allocation for effective maintenance strategies. According to the authors, the proposed DRL-based framework can be extended to other industries, such as healthcare, transportation, and energy, to improve maintenance practices and prevent catastrophic events. In conclusion, the study demonstrates the feasibility and effectiveness of using DRL for anticipating very uncommon failure events in aircraft maintenance. The framework includes a reward function that promotes the DRL algorithm to learn how to prioritize variables and allocate resources for maintenance tasks. The authors compare the proposed DRL performance approach with traditional machine learning models and show that the DRL approach outperforms these models in predicting rare failures.

In this article, we present an integrated machine learning model [14] for predicting rare aircraft component failures using log-based datasets. The proposed tactic combines multiple machine learning algorithms such as clustering, classification, and regression to predict the RUL of aircraft components and identify rare failure occurrences. The study uses a dataset containing maintenance logs of multiple aircraft components for the training and testing of an integrated machine learning model. The authors filter meaningful features, use clustering algorithms to group similar components, and then use classification algorithms to predict the probability of failures. Finally, they use regression algorithms to estimate each component's RUL and provide maintenance recommendations. The results indicate that the integrated machine learning tactic is superior to other models at making predictions about the RUL of aircraft components and identifying rare failures. According to the authors, the proposed model can be used to optimize maintenance practices and reduce the occurrence of catastrophic events in aviation.

A framework for prognosis and health management (PHM) [15] of aircraft is proposed in the study and uses a number of deep learning techniques. The suggested methodology seeks to determine the health state of aircraft components and anticipate their RUL based on sensor data. To extract features from sensor data and forecast the RUL of aeroplane components, the authors mix different deep learning methods, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and auto encoders. The proposed framework is assessed using a variety of performance metrics, including mean absolute error (MAE) and Root Mean Square Error (RMSE), and alternative machine learning models, including decision trees and Support Vector Machines (SVMs), are compared. The findings demonstrate that the suggested framework performs better in forecasting the RUL of aircraft components than the other models. The authors also propose an approach using auto encoders to learn the components' typical behaviour and spot irregularities in their sensor data. The authors evaluate the approach using a subset of the dataset and show that it can accurately detect anomalies and identify the health status of components.

The paper proposes an intelligent predictive maintenance approach [16] for aeroengines using a digital twin-based framework. A digital twin is a virtual replica of a physical system that simulates its behaviour in real-time and can be used for various applications, including predictive maintenance. The framework includes three main components: a digital twin, a health assessment module, and a predictive maintenance module. The study evaluates the proposed approach using a dataset containing sensor measurements of aeroengines. The authors train and test the proposed framework using the dataset and evaluate its performance in forecasting the RUL of aeroengines. The findings demonstrate that the suggested method may produce maintenance suggestions and properly anticipate the RUL of aeroengines. The authors also conducted a sensitivity analysis to assess the robustness of the suggested strategy under different levels of data noise and show that it is resilient to noise in the sensor data. The authors advise that the suggested strategy can potentially improve maintenance practices in aviation and recommend further research to optimize its performance.

Kernel parameter settings in the SVM training procedure have a significant impact on regression accuracy which is a step in this method of optimization [17]. The model used only needs to know the present state of the same data; it does not need to be aware of the aircraft engine's previous status. This approach offers the benefit of robustness of system against potential memory register failures. In order to reduce costs associated with RUL prediction, the Particle Swarm Optimization (PSO) method was effectively applied in this study to optimize the hyperparameters corresponding to the best SVM model for the RUL prediction from the other observed quality characteristics. This research has produced a hybrid PSO-RBF-SVM-based model. The findings show that the generalisation capability obtained with just the SVM-based regressor is considerably worse than that obtained with the PSO-SVM regression method. A coefficient of determination of 0.9034 was discovered using experimental data and this hybrid PSO-RBF-SVM-based model. This predictive model's lack of dependence on knowledge of the engine's previous operational states is one of its major advantages.

In this article [18], a comparative study of existing machine learning algorithms for predicting the remaining service life of aircraft turbofan engines is performed. The turbofan engine dataset from NASA's Prognostics Data Repository was used to create the machine learning model. The errors' root mean squares were displayed, and it was found that the random forest method produced the best outcomes. The random forest approach allows a large number of observations to participate in the prediction while simultaneously capturing the variation of several input variables. It was found that all ten algorithms consistently generated proportionate accuracy for the various algorithms put to the test.

The paper proposes a method for predicting [19] the remaining useful life (RUL) of machinery using an Artificial Bee Colony (ABC) algorithm optimized Echo State Network (ESN). The proposed method consists of two main steps. In the first step, the ESN is used to extract the relevant features from the raw sensor data. The extracted features are then used to predict the RUL of the machinery using the ABC algorithm in the second step. To optimize the hyper parameters of the ESN, the ABC algorithm is used improve the accuracy of the RUL prediction. The proposed method is validated using a turbofan engine dataset from the Prognostics Data Repository. The results show that the proposed method outperforms other existing methods for RUL prediction, even in terms of computational efficiency and real-time implementation. This method can potentially be applied to other datasets to predict the RUL of machinery and has potential for use in industrial settings.

The paper proposes a technique for predicting [20] the RUL of machinery using Recurrent Neural Networks (RNNs). The proposed method is based on a time-series analysis of the sensor data and the use of RNNs to predict the RUL. The paper compares the proposed RNN method with other existing methods for RUL prediction, including the Kalman filter and the support vector machine. The results depict that the RNN strategy outperforms the other methods in terms of accuracy, robustness. Handling missing data and addressing how the sensor data and RUL have a non-linear connection. The paper also discusses the practical issues associated with implementing the RNN method in an industrial setting, which requires vast amounts of training data, the challenge of selecting appropriate hyper parameters, and the potential for overfitting. The method has the potential to be used in industrial settings for predictive maintenance, however, more investigation is required to gauge its efficacy on other datasets and address practical implementation issues.

The paper [21] proposes a method for predicting the remaining useful life (RUL) of machinery using deep convolutional neural networks (DCNNs). The proposed method involves training the DCNNs on sensor data to predict the RUL of the machinery. The features are automatically extracted from the sensor data by the DCNNs and utilised to forecast the RUL. The method is evaluated using a dataset from the Prognostics Data Repository, and the results show that the proposed method outperforms other existing methods for RUL prediction, including long short-term memory networks, deep belief networks, and support vector regression. The paper also discusses the limitations of the proposed method, which require vast amounts of training data, the complexity of deciphering the DCNNs' output, and the potential for overfitting. The paper suggests several ways to address these limitations, including the use of transfer learning, visualization techniques, and regularization techniques. The proposed method has the potential to be used in industrial settings for predictive maintenance, but further research is needed to address the limitations of the method and evaluate its effectiveness on other datasets [22].

The proposed method [23] utilizes an improved version of the C-Loss Extreme Learning Machine (ELM) algorithm. The authors show that their method outperforms traditional machine learning techniques in terms of precision and computational effectiveness, including Support Vector Regression and Random Forest Regression. Furthermore, the authors demonstrate the practical usefulness of their approach by applying it to real-world scenarios. They provide an example of using their RUL prediction model to perform proactive maintenance on a fleet of aero engines, which can help reduce downtime and maintenance costs. Overall, the approach is said to have important implications for the aviation industry and offers a promising alternative to traditional machine learning techniques to help optimize maintenance and repair schedules for aero engine fleets.

The next step in developing an effective model for predicting the residual useful life of an aero engine was to combine more than one deep learning algorithm [24]. The proposed method uses the proposed method uses a combination of CNN and bidirectional Long Short-Term Memory (LSTM) networks to analyze time-series sensor data and predict possible machine failures. The proposed approach involves preprocessing raw sensor data using a CNN, which is then fed into a bi-directional LSTM network, to model the long-term temporal dependencies between the different sensor signals. This allows the network to effectively capture the complex dynamics of machine health, including both short-term and long-term changes in sensor signals. The authors demonstrate the success of their strategy using a dataset of vibration signals from rotating machinery, collected under different operating conditions and with different fault types. They compare their proposed method with various other machine learning techniques that include decision trees, support vector machines (SVM), and random forests. According to the findings, the suggested strategy performs better in terms of accuracy, sensitivity, and specificity than these other methods. They also evaluate the robustness of their proposed approach to noisy data, by introducing various levels of noise to the vibration signals. The results depict that the proposed tactic is more robust to noise than the other machine learning techniques.

Another approach based on deep learning for predicting the remaining useful life of aircraft engines can be seen in [25]. It proposes a novel approach for predicting the RUL of engineered systems using vanilla LSTM neural networks. An additional aim is to address the issues of RUL estimation in engineered systems, such as limited data availability and the need for accurate and timely predictions. The proposed approach involves using a vanilla LSTM network to model the RUL of technical systems using standard LSTM networks. The authors evaluate the effectiveness of their proposed approach using a dataset of engine RUL, which includes sensor measurements and maintenance records. They compare the performance of their method to several other machine learning strategies, such as Support Vector Regression (SVR) and Random Forest Regression (RFR).

The findings show that the proposed strategy outperforms the other machine learning techniques with regards to accuracy and computational effectiveness. They note that the proposed method may not be suitable for all types of engineered systems, and that additional data pre-processing and feature engineering may be necessary in some cases. They also suggest that further research could explore the use of more complex LSTM networks, as well as the integration of other types of data. The outcomes of a case study show how helpful this technique is in practise for real-world applications and hint that it can be a useful tool for enhancing the safety and dependability of engineered systems across a range of industries.

The Table 1 shows the summary of the different methodologies.

3. DESIGN AND IMPLEMENTATION

3.1. Dataset definition and processing

The data for this project has been taken from a publicly available dataset repository hosted by NASA's Prognostics Centre (National Aeronautics and Space Administration). This is a collection of datasets donated by universities, institutions, and companies. The data repository attentions exclusively on forecast datasets, i.e., datasets that can be used to develop forecasting algorithms. Most of these are time series data from the previous normal state to the failed state.

A tool for simulating realistic data for big commercial turbofan engines is the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) was used to simulate engine degradation during the dataset collection process. This records multiple sensor channels to track the evolution of faults. Every flight is made up of a number of different flight circumstances, each with a sufficient linear transition time that allows the engine to switch from one to the next. The parameters for each flight are the flight circumstances, health indicators, measurement temperatures, and pressure measurements. A flight is a complete flight recording sampled at 1 Hz. There were problems with the intake engine fan, the high-pressure turbine, low pressure turbine, low pressure compressor, and high-pressure compressor.

The dataset is organized as csv files having the following columns - unit number (to denote the flight engine being sampled), cycles (to indicate the flight cycle number for which the data is provided), 3 operational settings (which have an impact on the engine performance), 21 sensor measurements hosting a variety of temperature measurements, pressure measures, and health indicators.

The concepts of variance threshold and correlation coefficient analysis are used to select features; for the former, we eliminate parameters whose values appear to have a nearly constant trend across all flight cycles of operation, and for the latter, we select any one column from a pair of closely correlated parameters. As seen in Figure 1, for the sensor measurements, we plotted the values of their standard deviation across all the rows of the dataset. Values that were extremely close to the x-axis were discarded. Thus, here we discarded sensor measurements 1, 5, 6, 16, 18 and 19.

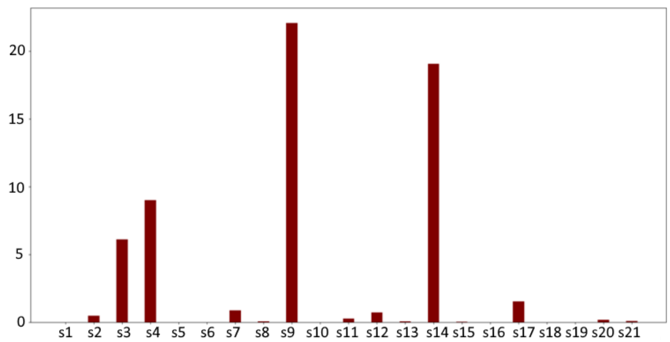

Figure 1. Plot of standard deviations of sensor values for feature selection.

Table 1. Summary of the different methodologies.

Paper Reference	Key Focus	Methodology and Algorithms	Results	Recommendations and Conclusions
[1]	Predictive Maintenance using GOA and ESN	Train ESN on engine vibration data; Optimize using GOA	Achieved 97.9 % prediction accuracy; Outperformed BPNN, SVM, DT	Recommends the proposed approach for real-world applications
[2]	Remaining Usable Life Prediction using CNN and LSTM	Develops CLSTM model for estimating engine's remaining usable life	Successful learning of historical data; Future scope includes energy-efficient approach	
[3]	Trends and Challenges in Predictive Maintenance	Review of ML approaches; Challenges in aircraft maintenance	Recommends hybrid models, additional data sources, and standardized frameworks	
[4]	Feature Selection for Prognostics and Health Management	Systematic methodology for PHM system needs; Case study	Guidelines for PHM system development; Applicable to other aviation systems	
[5]	Aircraft Hydraulic System Monitoring and LSTM Prognostics	Machine learning for condition monitoring; LSTM for prognostics	LSTM-based model accurately predicts engine behaviour	Overview of aircraft hydraulic systems and predictive maintenance
[6]	Dynamic Maintenance Architecture using CNN	CNN for RUL prognostics; ILP for maintenance scheduling	CNN-based RUL prognostics evaluated using RMSE metric	Suggested maintenance planning framework for aircraft components
[7]	State-of-the-Art Overview of Predictive Maintenance	Overview of projects and challenges; Optimization of PdM	Challenges include imbalanced datasets and high dimensionality	Optimization of PdM over alternative maintenance strategies
[8]	Operational Factors and Reliability Estimates	Data-driven strategy using PHMs; Improved maintenance schedules	Improved reliability estimates decrease unscheduled occurrences	Methodology applied to decrease unscheduled occurrences
[9]	DRL for Maintenance Planning	DRL strategy for maintenance planning; Improved overall cost	DRL strategy decreases overall maintenance cost by 29.3 %	Efficient maintenance planning based on probabilistic RUL prognostics
[10]	RUL Prediction with Ambiguity and Decision-making	Bi-LSTM for RUL prediction with ambiguity; Maintenance cost function	Method considers uncertainty and maintenance-related costs	Adaptive maintenance timing based on economic requirements
[11]	GRU for Preventive Maintenance	GRU for RUL prediction; Faster prediction process	GRU method has faster prediction process and simpler calculation	Challenges in shortening LSTM training time while maintaining accuracy
[12]	Improved SVM Model for RUL Prediction	SVM model with hybrid PSO-RBF-SVM; Outperforms VARMA	Proposed model achieves high coefficient of determination	SVM model outperforms VARMA model
[13]	DRL Framework for Anticipating Rare Failure Events	DRL-based framework for rare failure events; Outperforms traditional models	DRL approach outperforms traditional models in predicting rare failures	DRL-based approach applicable to various industries
[14]	Integrated Machine Learning Model for RUL Prediction	Clustering, classification, and regression for RUL prediction	Integrated model superior in predicting RUL and identifying rare failures	Model optimization for maintenance practices in aviation
[15]	PHM Framework using CNNs, RNNs, and Autoencoders	Framework for PHM using deep learning techniques	Proposed framework outperforms other models in forecasting RUL	Autoencoders for anomaly detection in sensor data
[16]	Digital Twin-based Predictive Maintenance	Predictive maintenance using digital twin; Robust to noise	Proposed approach produces maintenance suggestions and accurately predicts RUL	Potential improvement of maintenance practices in aviation
[17]	SVM Training with PSO Optimization	PSO optimization for SVM regression parameters	Hybrid PSO-RBF-SVM-based model outperforms SVM-based regressor	Robust regression accuracy with PSO optimization
[18]	Comparative Study of ML Algorithms for Turbofan Engines	Comparative study for RUL prediction; Random Forest tops	Random Forest method produces best outcomes; Consistent accuracy among methods	Comparative analysis of machine learning algorithms for RUL prediction
[19]	ABC-Optimized ESN for RUL Prediction	ABC algorithm for optimizing ESN; Outperforms existing methods	Proposed method outperforms existing methods in RUL prediction	Real-time implementation potential and computational efficiency
[20]	RNN-based RUL Prediction	Time-series analysis using RNNs; Outperforms other methods	RNN strategy excels in accuracy, robustness, and handling missing data	Practical challenges include data availability and hyperparameter selection
[21]	DCNNs for RUL Prediction	Predicting RUL using DCNNs; Outperforms other DL models	DCNNs outperform LSTM, DBN, and SVR in RUL prediction	Limitations include vast training data requirement and complexity
[22]	C-Loss ELM Algorithm for Improved Predictions	Improved C-Loss ELM algorithm for precision and efficiency	Outperforms traditional ML techniques like SVR and Random Forest	Practical usefulness demonstrated in proactive maintenance
[23]	LSTM and CNN Combination for RUL Prediction	CNN for preprocessing and LSTM for modelling temporal dependencies	Proposed method outperforms decision trees, SVM, and random forests	More robust to noise than other machine learning techniques
[24]	Combining CNN and Bidirectional LSTM for Machinery Health	Combination for analysing time-series sensor data; Outperforms other methods	Robust to noisy data; Superior accuracy, sensitivity, and specificity	Potential for real-world application in predicting machine failures

Upon generating the correlation graph between the parameters, it was observed that sensor measurements 9 and 14 were highly correlated, so the decision was taken to drop column 14 and retain column 9. For the operational settings, it was observed that operational setting 3 had a standard deviation of 0.0, so it was discarded as well.

Lastly, a column for the remaining useful life, which will serve as the target variable for the supervised learning model, had to be included in the training dataset for each aircraft cycle, and this was obtained by subtracting the maximum cycle number for that specific unit from the current cycle number.

The proposed model's architecture model is depicted in Figure 2. Collecting and preparing the data is the first and most crucial step of machine learning. It involves selecting the right data sources, cleaning, and transforming the data to ensure its quality, and preparing it for the model. The next step is to transform the data into a format that can be easily analysed and used to train the model. This includes techniques such as data normalization.

The next step involves selecting and extracting the applicable characteristics from the data that can help in predicting the outcome. Hyper parameter tuning is the next step in the workflow that involves fine-tuning the hyper parameters of the model to achieve better performance. The best collection of hyper parameters must be chosen for hyper parameter adjustment that minimize the error or maximize the accuracy of the model. This is done by trying out different combinations of hyper parameters and evaluating the performance of the model on a validation set. Hyper parameter tuning is an iterative process that may require several rounds of experimentation and evaluation until the optimal set of hyper parameters is found. Metaheuristic algorithms are optimization algorithms that can efficiently search through large and complex search spaces to identify the ideal combination of hyperparameters. They are particularly useful for hyper parameter tuning because they can search through a huge number of hyper parameters efficiently and effectively, without being trapped in the local optima and can optimize the hyper parameters of a complex machine learning model without requiring knowledge of its internal workings or the underlying data distribution. The next step involves using the selected model to train on the prepared data using the most optimal parameters obtained from the last step. The next step involves evaluating the trained model's performance on a test set of data. Performance metrics for the model include accuracy, precision, recall, and F1 score.

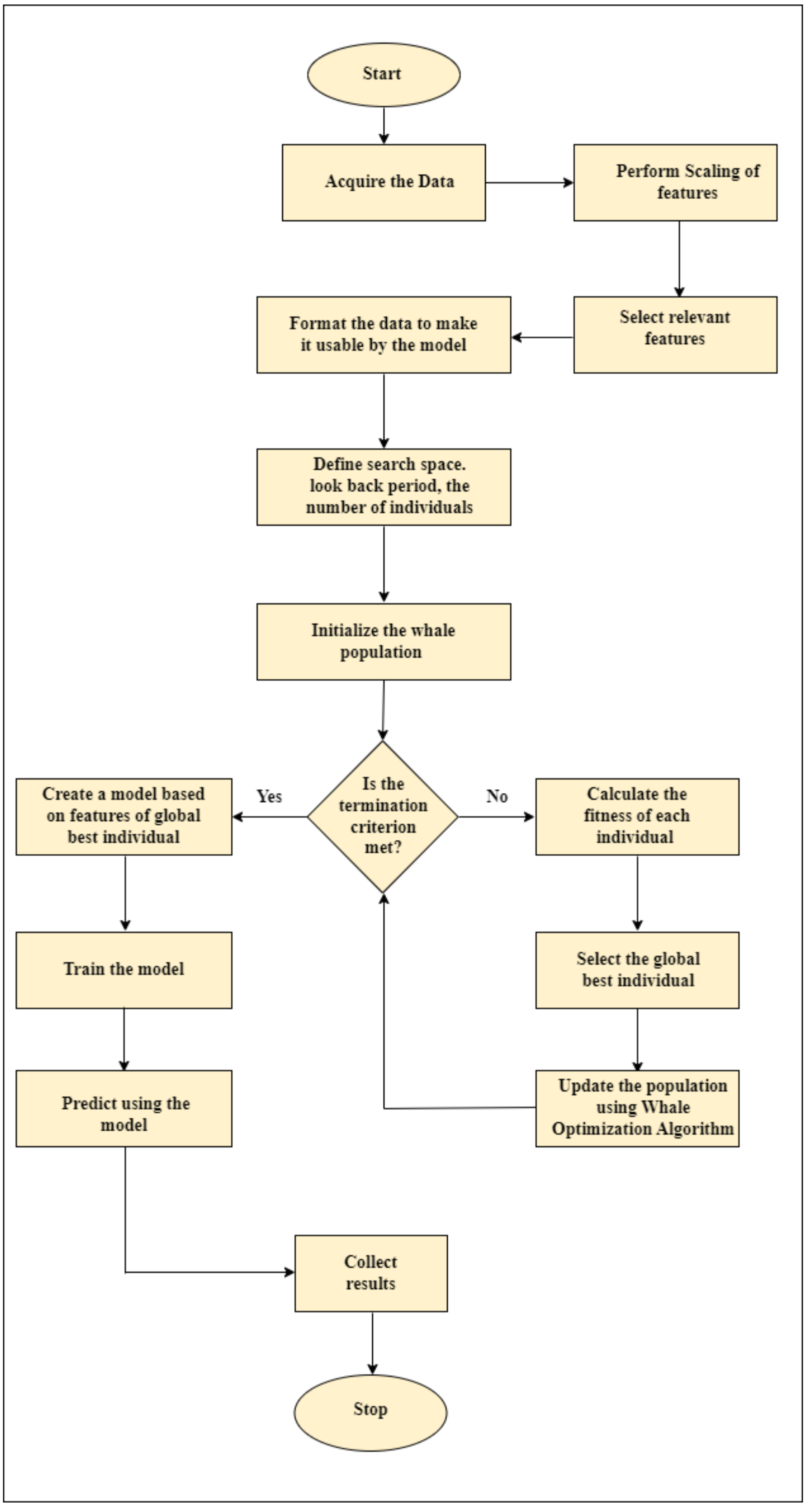

Figure 2. Architecture of the proposed model.

3.2. Whale Optimization Algorithm

Whales are mammals that can reach to 30 m and 180 tonnes in height and weight, respectively. There are multiple species of this fascinating creature but the species of interest in this algorithm are called Humpback whales. Whales are regarded as highly emotional and clever mammals that usually forage in communities. The hunting strategy of Humpback whales is noteworthy, referred to as bubble-net feeding method, which was only observed from the surface of oceans for a long time. Summarising their hunting behaviour, the whales dive to a depth of about 12 metres and begin making bubbles around the prey in a spiral shape, thus tricking the fish into thinking they are trapped within a cylindrical column. This column goes on shrinking in size until the prey is confined to a spot and the whales can consume it. The pseudocode for the same can be seen in Pseudocode 1.

Pseudocode 1: WOA

```
Input: Xi
Output: X*
Initialize the population Xi(i= 1,2,…n)
Calculate the fitness of each search agent
X* = the best search agent
while(t<maximum_number_of_iterations)
   for each search agent
   Update a,A,C,l,p
   if(p<0.5)
      if(|A|<1)
         Update the position of the current
         search agent
      else if(|A|>1)
         Select a random search agent
         (Xrand)
         Update the position of the current search agent
      end if
   else if(p>0.5)
      Update the position of the current search agent
   end if
   end for
   Check if any search agent goes beyond the search space and adjust
   it
   Calculate the fitness of each search agent
   Update X* if there is a better solution
   t = t+1
end while
return X*
```

3.3. Binary Classification of Engine Flight Capability

a) LSTM model without hybridization

The first task that the project aims to tackle is that of binary classification of an engine's capability to take flight. This is decided based on the parameter of engine's RUL of the engine when compared to a fixed value. The RUL of an aircraft engine refers to the time or use that an engine has left before it needs to be replaced or overhauled. RUL estimation is an important task in aircraft maintenance, as it helps to predict potential engine failures and schedule maintenance activities more efficiently.

If RUL is greater than the specified value, the engine is considered airworthy; otherwise, it is classified as unflyable. This task of binary classification has been addressed with the help of an LSTM model given its strength in the field of analysing time series data similar the dataset considered for this study.

The first step in this process is the selection of a random seed. The random seed in an LSTM network determines the starting point of the random number generator used during the training of the model. By setting a fixed random seed, the LSTM model will generate the same sequence of random numbers each time it is trained, ensuring that the results are reproducible.

Setting a random seed is particularly important when conducting experiments or comparing different models, as it allows researchers to ensure that any differences in performance are due to the differences in the models themselves, and not due to chance variations in the random initialization of the model.

This is followed by loading the training, testing, and ground truth datasets into the environment. The datasets are such that the columns aren't annotated. Thus, the first pre-processing step would be to annotate the dataset. The dataset is then sorted to make the analysis easier. The dataset is sorted using the 'id' column, which is a unique identifier assigned to every engine in the dataset, followed by a sort based on 'cycle' column. The cycle column is an indicator of which flight cycle's sensor values have been recorded in that particular row. This is followed by a feature selection process and the addition of an RUL column.

Following this procedure for data preprocessing, we create two additional labels called 'label1' and 'label2' with threshold values of w1 and w2, both of which dictate the limits of the RUL values to determine the viability of an engine to continue taking flights.

The data must be normalized for smooth analysis of the dataset, using the MinMaxScaler normalizer found in Keras. For the testing dataset, the ground truth values are available, hence, they are merged with it to create a new column called 'max' for each 'id' which holds the value of the maximum possible flights the engine can take before it is decommissioned. The same procedure is repeated with the testing dataset as well. The 'label1' and 'label2' columns created before for the training set are also created for the testing set to determine the limits of RUL values to determine the viability of an engine to continue taking flights.

Following this the training and testing data frames are separated into the features and target columns. They are reshaped into a 3-dimensional array consisting of - the number of time steps, the features and the samples.

The model design chosen here was one consisting of 4 LSTM units, 1 Dense unit with a sigmoid activation function, and it was created using an Adam optimizer and a binary cross-entropy loss function. This generated 325 trainable parameters. The sequence length chosen was 50. The performance indicators identified in this study are accuracy, recall, accuracy, F1 score, and other performance indicators. If the RUL is greater than the specified value, the engine is considered efficient; otherwise, it is classified as unflyable.

This task of binary classification has been addressed with the help of an LSTM model, given its strength in the field of analyzing time series data similar to the dataset considered for this study. An algorithm for the same is given in Figure 3.

The first task that the project aims to tackle is the binary classification of an engine's capability to take flight. This is decided based on the residual useful life (RUL) of the engine when compared to a fixed value. When RUL exceeds the specified value, then the engine is considered capable; otherwise, it is classified as incapable of flying.

This task of binary classification has been addressed with the help of an LSTM model given its strength in the field of analyzing time series data similar to the dataset considered for this study in Pseudo Code 2.

Pseudo Code 2: LSTM algorithm for binary classification of Engine Flight Capability

```
Input: Training Datasets
Output: RUL Values
Perform feature selection
Load the training and test datasets, group by UnitNumber, add a
RUL column.
Create dataframes for training and testing as well as a dataframe of
true RULs for testing units.
Scale the features in the training and testing datasets.
Define the sequence length and feature columns to be used in
generating the training and test data.
Prepare the training data into a 3D numpy array with the shape of
(samples, time steps, features).
Prepare the training labels into a 1D numpy array.
Prepare the test data into a 3D numpy array with the shape of
(samples, time steps, features).
Define a Sequential model with LSTM layer(s), followed by Dense
layer(s) and Activation layer(s).
Compile the model with RMSprop or Adam optimizer.
Fit the built-in model using batch size and epochs to the training set
of data.
```

12 Using the evaluate() method, assess the model against the test set of data.
13 Apply the predict() procedure to each unit in the test data set to determine the RUL.
14 Return the predicted RUL values in a pandas DataFrame with corresponding unit numbers.

b) LSTM model with hybridization

The previous approach of just using LSTM alone for the purpose of deciding the viability of an engine was not up to the required accuracy levels. Thus, to further improve the performance of this classification model it was identified that the task of hyperparameter tuning was necessary.

Hyperparameter tuning is an essential part of developing machine learning models. Hyperparameters are values that are set before training a model, and they affect the behaviour of the model during training. For instance, the hyperparameters in a LSTM model, the hyperparameters could include the number of covert units, learning rate, dropout rate, the number of epochs, and so on.

Finding the best hyperparameters for an LSTM model can be challenging because there are often many different hyperparameters to optimize, and the search space can be vast. Grid search and random search are two traditional methods of hyperparameter tuning. Grid search involves defining a set of hyperparameters to be tested, and then training the model for each combination of hyperparameters. This method can be time-consuming, and it becomes computationally expensive for high-dimensional search spaces. Random search, on the other hand, selects hyperparameters at random from a given distribution. This method is less computationally expensive than grid search, but it can still be time-consuming, especially when searching for optimal hyperparameters.

Metaheuristic algorithms are a family of optimization algorithms that can be used to solve complex optimization problems like hyperparameter tuning. These algorithms are inspired by natural processes such as evolution, swarm behaviour, and annealing. They are designed to explore the search space efficiently and find optimal solutions.

One significant advantage of using metaheuristic algorithms for hyperparameter tuning of LSTM models is their ability to explore the search space efficiently. In contrast to traditional methods such as grid search and random search, metaheuristic algorithms can search the hyperparameter space more intelligently. This is because they use intelligent search strategies that can quickly find promising areas of the search space.

Another benefit of using metaheuristic algorithms for hyperparameter tuning of LSTM models is improved optimization. Metaheuristic algorithms are designed to avoid getting stuck in local optima. In contrast, traditional methods like grid search and random search can get trapped in local optima, which can result in suboptimal solutions.

The Whale Optimisation Algorithm (WOA) is a metaheuristic optimization algorithm that draws inspiration from humpback whales' hunting strategies. It has been used for various optimization problems, including hyperparameter tuning for machine learning models like deep neural networks [26], [27].

In our use case, we have decided to tune the following hyper parameters: LSTM layers, wherein they take a value between 1 and 3, LSTM units per layer, wherein they can take any value between 50 and 100, dropout-bounds, wherein any value for the dropout rate between 0 and 0.5 is chosen: and learning rate bounds - wherein any value between 0.000001 and 0.001 is preferred for the learning rate.

Data preparation for the model is done in a similarly to the previous LSTM model itself. The modification manifests itself in the introduction of the Whale Optimization Algorithm to find the optimal set of hyper parameters. Initially a whale population is initialised with every whale being an n-membered tuple holding random values for the n hyper parameters chosen, such that the values are within the bounds defined in the search space. The zeroth whale is chosen as the global best whale in the zeroth iteration to start off the optimization process. An LSTM model is then created for every set of hyperparameters and trained and validated on the dataset. The concept of Early Stopping is incorporated into the model to ensure the efficient utilisation of both time and computational resources. The fitness function chosen here is one to measure which whale has the lowest value of binary-cross-entropy loss. Naturally, the set of hyper parameters that gives the lowest value of binary-cross-entropy loss, is the best performing agent of the lot. With the updating of the whales' positions, their values are then updated in the direction of the best agent. After updating, a check is done to ensure that the hyper parameter values are within the bounding ranges defined initially. This optimization process is repeated over a fixed number of iterations to finally obtain the global best agent whose fitness value, which is its binary cross entropy loss value, is called best fitness and whose hyperparameters are called the best parameters as in Pseudo Code 3.

Pseudo Code 3: WOA+LSTM algorithm for binary classification of Engine Flight Capability

Input: Training Dataset
Output: Binary Classification for LSTM model

1 Set the random seed.
2 Load training, testing and ground truth datasets and annotate column names.
3 Sort training data by the 'id' and 'cycle' columns.
4 Perform feature selection and add RUL column.
5 Create label1 and label2 columns with threshold values 'w1' and 'w0'.
5 ***if*** RUL <= W1
6 label as 1
7 ***else***
8 label as 0.
9 ***if*** RUL <= W0
10 label as 2.
11 Normalize 'cycle' column in training and testing data frames
12 Merge ground truth with the testing data frame to create new column 'max' for each 'id'
13 Merge training with new ground truth data frame based on 'id' column to add 'RUL' to the testing data frame.
14 Create two new label columns 'label1' and 'label2' in the testing data frame based on the same threshold values as the training data frame.
15 Separate the features and target columns from the training and testing data frames.
16 Reshape the training and testing features into a 3D array of shape.
17 Define a binary classification LSTM model.
18 Compile the model using binary cross-entropy loss and the Adam optimizer.
19 Fit and evaluate the model on necessary metrics

3.4. Remaining Useful Life Prediction

a) LSTM model without hybridization

The second task that this paper will deal with is the prediction of the RUL of an aircraft engine. RUL is a measure of the number of flights an engine can take given its current state. This metric depends on a number of factors, like the combination of sensor values or even operating conditions. Thus, it is the job of the deep learning model to be able to effectively predict the RUL of a flight engine given the collection of sensor measurements and operating conditions.

The first step in this process is performing feature selection. We have employed a procedure similar to the Binary Classification Model. This is followed by a scaling of the sensor values in the dataset. The features are scaled using the MinMaxScaler. MinMaxScaler is a data normalization technique commonly used in machine learning to scale characteristics of input data to a specific range.

In Keras, the MinMaxScaler is implemented in the pre-processing module, and it can be used to scale input features ranges 0 and 1. This is accomplished by subtracting the minimum value of the feature and dividing it by the range (i.e., the difference between the maximum and minimum values). This scaling technique is especially useful for tactics that are sensitive to the scale of the input features, such as neural networks. By scaling the features, the model can converge faster and may achieve better accuracy.

A sequence length of 50 is defined for this model to ensure enough prior information is used to predict the value of the residual useful life of the engine in the test set. The training data is prepared into the 3-dimensional arrays consisting of features, time steps, and samples. The training labels, which are the RUL values, are ordered into a 1-dimensional array. This is followed by making the test set into a similar 3-dimensional array consisting of features, time steps, and samples. A Sequential model is defined for this problem statement consisting of an appropriate number of LSTM layers, followed by Dense and Activation Layers.

The model is compiled using RMSProp and Adam optimizer function. RMSProp is an optimization algorithm commonly used in deep learning to update a neural network's weights during training. It stands for Root Mean Square Propagation and is designed to reduce the impact of large gradients in the optimization process, which can lead to slow convergence or divergence of the model. RMSProp calculates a moving average of the squared gradients and uses this to normalize the learning rate for each weight. This normalization helps to prevent oscillations in the optimization process and allows for faster convergence. RMSProp is particularly useful for models with sparse gradients, such as those with ReLU activation functions, and it is commonly used in conjunction with other optimization techniques such as Adam.

This compiled model is then fit to the training data and evaluated on the test dataset to predict the RUL values for the same. These RUL values are then returned in a pandas Dataframe for better readability. Each predicted RUL value is affixed with the corresponding unit number to make the results more legible.

The algorithm for this approach using just an LSTM model is outlined in Pseudo Code 4.

Pseudo Code 4: LSTM model for RUL prediction

Input: Training Datasets
Output: Model with Adam Optimizer

1 Perform feature selection
2 Load the training and test datasets, group by UnitNumber, add a RUL column.
3 Create dataframes for training and testing as well as a dataframe of true RULs for testing units.
4 Scale the features in the training and testing datasets.
5 Define the sequence length and feature columns to be used in generating the training and test data.
6 Prepare the training data into a 3D numpy array with the shape of (samples, time steps, features).
7 Prepare the training labels into a 1D numpy array.
8 Prepare the test data into a 3D numpy array with the shape of (samples, time steps, features).
9 Define a Sequential model with LSTM layer(s), followed by Dense layer(s) and Activation layer(s).
10 Compile the model with RMSprop or Adam optimizer.
11 Fit the built-in model using batch size and epochs to the training set of data.
12 Evaluate the model on the test data set using evaluate () method.
13 Utilizing the predict () approach, forecast the RUL for each unit in the test data set.
14 Return the predicted RUL values in a pandas Data Frame with corresponding unit numbers.

b) LSTM model with hybridization

The previous strategy of solely using LSTM to assess an engine's viability did not achieve the necessary levels of accuracy. Thus, it was determined that hyperparameter tuning was required to further enhance the performance of this classification model. To create a hybrid entity capable of effectively carrying out the task at hand, the Whale Optimization Algorithm was combined with the designed LSTM model. Metaheuristic algorithms have shown promise in this task for previous problem statements, which is why this decision was made. In Figure 7.6, the algorithm for this strategy is highlighted.

The scaling of features in the training and testing datasets is carried out using the Min/Max Scaler function. The LSTM model hyperparameters include the sequence length, number of LSTM units, number of dense units, dropout rate, learning rate, and number of epochs. The LSTM model is defined using Keras Sequential and adds a masking layer, a LSTM layer with the provided number of units and a linear activation function, an LSTM layer with the specified number of units and a dropout rate, and a final dense layer with a single unit.

Furthermore, fitness function for WOA takes in the LSTM model, the training and testing dataframes, and the hyperparameters and returns the RMSE for the predicted RULs.

The data preparation for the model is done in a manner similar to the previous LSTM model itself. The modification manifests itself in the introduction of the Whale Optimization Algorithm to find the optimal set of hyper parameters. Initially a whale population is initialised with every whale being an n-membered tuple holding random values for the n hyper parameters chosen, such that the values are within the bounds defined in the search space. The zeroth whale is chosen as the global best whale in the zeroth iteration to start off the optimization process. An LSTM model is then created for every set of hyper parameters and trained and validated on the dataset. The concept of Early Stopping is incorporated into the model to ensure the efficient utilisation of both time and computational resources. The fitness function chosen here is one to measure which whale has the lowest MSE loss value. Naturally, the set of hyper parameters which gives the lowest value of MSE loss value is the best performing agent of the lot. When updating the whales' position, their values are then updated in the direction of the best agent. After updating, a check is done to ensure that the hyper parameter values are within the bounding ranges defined initially. Over a predetermined number of iterations, this optimization process is repeated and then finally gives the global best agent whose fitness value which is its MSE loss value is called best fitness, and hyper parameters are called best parameters.

The previous strategy of solely using LSTM to assess an engine's viability did not achieve the necessary levels of accuracy. Thus, it was determined that hyper parameter tuning was required to further enhance the performance of this classification model. To create a hybrid entity [28] capable of effectively carrying out the task at hand, the Whale Optimization Algorithm was combined with the designed LSTM model. Metaheuristic algorithms have shown promise in this task for previous problem statements, which is why this decision was made. In Pseudo Code 5, the algorithm for this strategy is highlighted.

The MinMaxScaler function is used to scale the features in the training and testing datasets. The LSTM model hyperparameters include the sequence length, number of LSTM units, number of dense units, dropout rate, learning rate, and number of epochs. The LSTM model is defined using keras Sequential and adds a masking layer, a LSTM layer with the provided number of units and a linear activation function, an LSTM layer with the specified number of units and a dropout rate, and a final dense layer with a single unit.

Furthermore, fitness function for WOA takes in the LSTM model, the training and testing data frames, and the hyper parameters and returns the RMSE for the predicted RULs.

Pseudo Code 5: LSTM+WOA model for Remaining Useful Life Prediction

Input: Training Datasets
Output: Whale optimized LSTM model

1. Perform feature selection
2. Load the training and test datasets, group by UnitNumber, add a RUL column.
3. Create dataframes for training and testing as well as a dataframe of true RULs for testing units.
4. Scale the features in the training and testing datasets.
5. Define the sequence length and feature columns to be used in generating the training and test data.
6. Prepare the training data into a 3D numpy array with the shape of (samples, time steps, features).
7. Prepare the training labels into a 1D numpy array.
8. Prepare the test data into a 3D numpy array with the shape of (samples, time steps, features).
9. Define the hyperparameters for the LSTM model.
10. Define the LSTM model.
11. Compile the LSTM model using the RMSprop optimizer and learning rate that have been selected.
12. Define the fitness function for WOA.
13. The hyperparameters of the LSTM model should be optimised using the WOA algorithm.
14. Train the LSTM model using the optimized hyperparameters and early stopping.
15. Predict the RULs for each unit using the trained LSTM model.
16. Calculate the RMSE and plot the predicted RULs against the true RULs.

The computational complexity of a Long Short-Term Memory (LSTM) model can be analysed based on the operations performed in each time step. The complexity includes Input Transformation, Memory cell updates. Considering all these components together, the overall computational complexity per time step for an LSTM unit is roughly O (dm + 4dm + 4m^2 + do). In practice, the dominating factors are often the matrix multiplications involving the input and hidden state.

The Whale Optimization Algorithm (WOA) is a metaheuristic optimization algorithm inspired by the social behaviour of humpback whales.

The optimization technique reduced the computational complexity of the LSTM model.

4. RESULTS AND INFERENCES

4.1 Binary Classification of Engine Flight Capability

Several deep learning techniques to forecast the binary classification of engine flight capability have been utilized, and model performance has been assessed through the use of accuracy and loss metrics. The results were then visualized through plotted graphs, which were compared and contrasted to determine the most effective approach. These graphs offer insight into the potential of deep learning methods for predicting engine flight capability, and the importance of choosing the best approach for a certain dataset.

1) LSTM WOA

The graph in Figure 3 displays the accuracy of a model that utilizes LSTM hybridized with WOA to predict the binary classification of engine flight capability. The accuracy of the model is shown on the y-axis, while the x-axis indicates the number of epochs. The model achieved an impressive 98.32 % accuracy by the end of the training process, demonstrating the efficacy of this approach for accurately forecasting engine flight capability.

The graph presented in Figure 4 depicts the loss of a model that utilizes LSTM hybridized with WOA to predict the binary classification of engine flight capability. The number of epochs

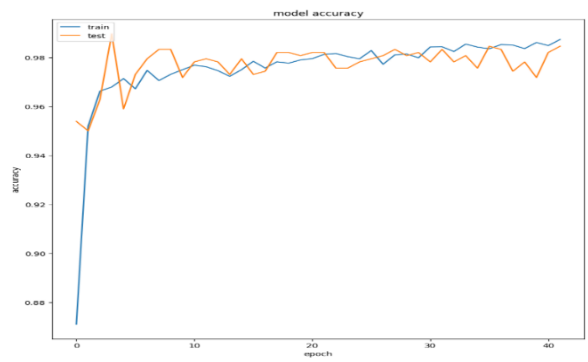

Figure 3. LSTM + WOA Accuracy graph.

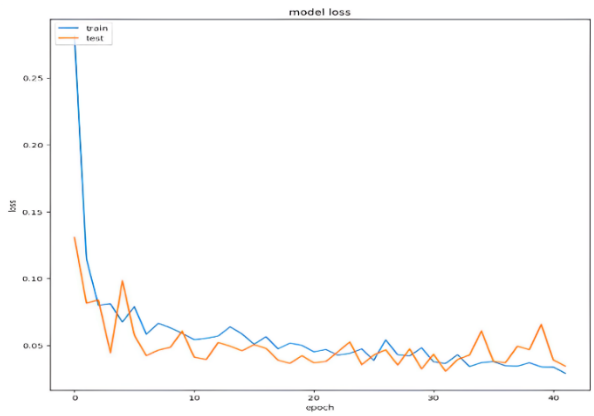

Figure 4. LSTM + WOA loss graph.

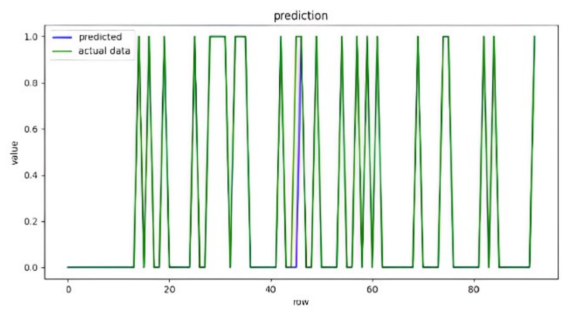

Figure 5. LSTM + WOA actual and predicted values graph.

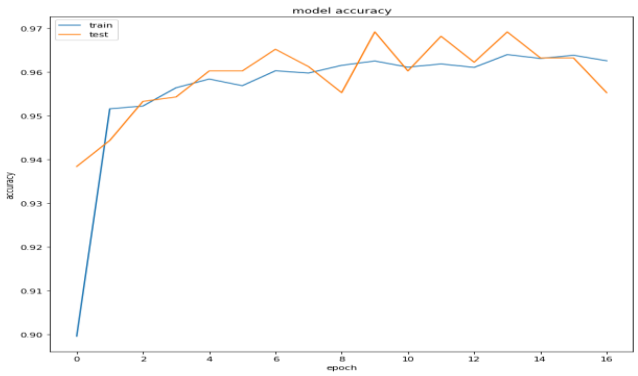

Figure 6. LSTM accuracy graph.

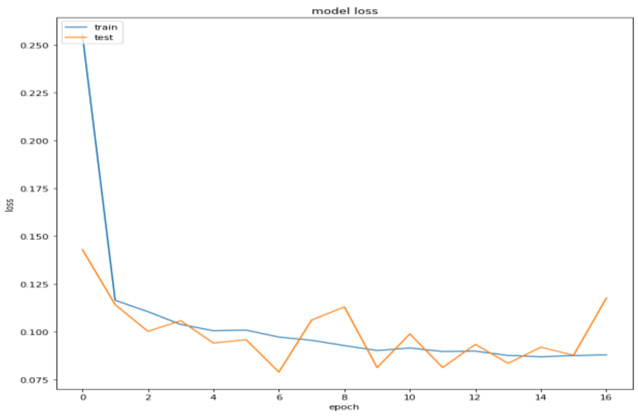

Figure 7. LSTM loss graph.

is shown on the x-axis, and the y-axis shows the model loss. The model achieved a final loss of 0.032, indicating that it was successful in minimizing the distinction between expected and actual values. This graph highlights the effectiveness of the LSTM hybridized with WOA approach for accurately predicting engine flight capability.

The graph in Figure 5 plots the actual and predicted values of a model that utilizes LSTM hybridized with WOA to predict the binary classification of engine flight capability. The graph displays the degree to which the projected and actual values match, providing insight into the accuracy and efficacy of the model.

2) LSTM

The graph in Figure 6 displays the accuracy of a model that utilizes LSTM to predict the binary classification of engine flight capability. The model's accuracy is shown by the y-axis, while the number of epochs is represented by the x-axis. The model achieved 95.53 % accuracy by the end of the training process.

The graph presented in Figure 7 depicts the loss of a model that utilizes LSTM h to predict the binary classification of engine flight capability. The number of epochs is shown on the x-axis, and the y-axis shows the model loss. The model achieved a final loss of 0.1176.

The binary classification of engine flight capability is predicted using an LSTM model, and the graph shown in Figure 8 shows the actual and anticipated values of the model. Unfortunately, the graph suggests that there was a significant gap between the anticipated and real values, indicating that the model did not do well in properly predicting the values. This graph highlights the limitations of the LSTM model in this particular

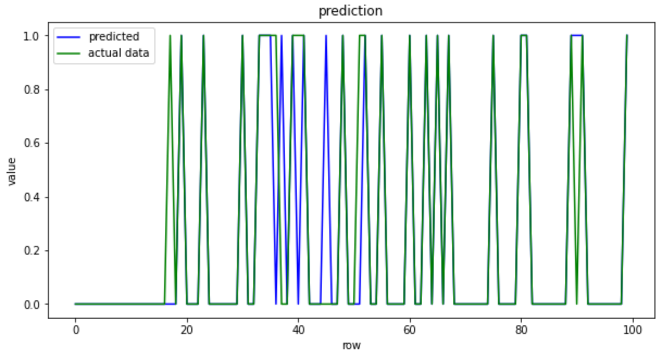

Figure 8. LSTM actual and predicted values graph.

application and indicates the need for further development and exploration of other modelling techniques.

3) 1D- CNN

The accuracy of a model that applies 1D-CNN to predict the binary classification of engine flight capability is shown in the graph in Figure 9. The y-axis displays the model's accuracy, and the x-axis displays the number of epochs. At the conclusion of the training procedure, the model had a 94 % accuracy rate.

The graph presented in Figure 9 depicts the loss of a model that utilizes CNN to predict the binary classification of engine flight capability. The number of epochs is shown on the x-axis, and the y-axis shows the model loss. The model achieved a final loss of 0.5738.

The graph presented in Figure 10 depicts the actual and predicted values of a CNN model used to predict the binary classification of engine flight capability. Unfortunately, the graph suggests that the model didn't work very well. In accurately predicting the values, with a large discrepancy between the predicted and actual values. This graph highlights the limitations of the CNN model in this particular application and indicates the

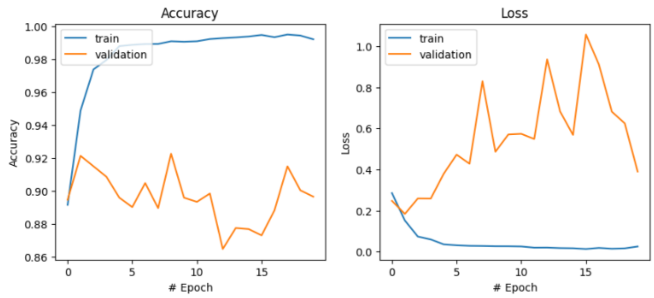

Figure 9. 1D - CNN accuracy graph and 1D - CNN loss graph.

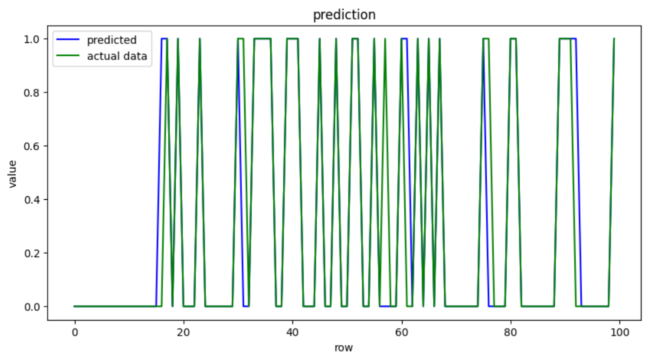

Figure 10. 1D - CNN actual and predicted values graph.

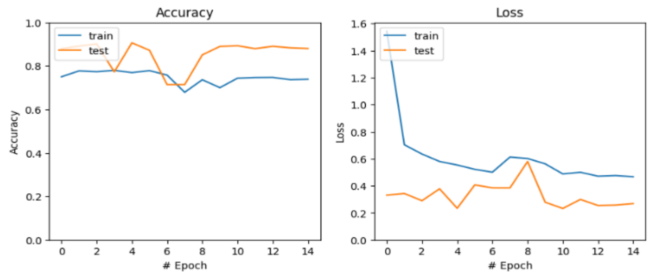

Figure 11. Deep CNN accuracy graph and Deep CNN loss graph.

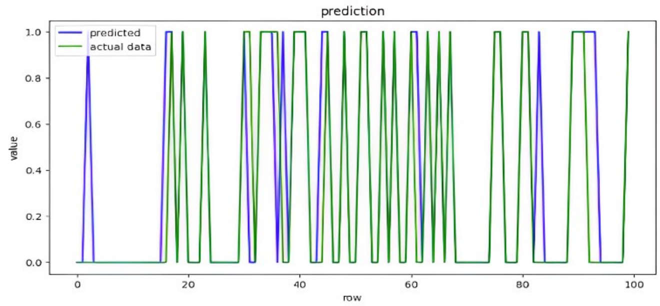

Figure 12. Deep CNN actual and predicted values graph.

need for further development and exploration of other modelling techniques.

4) Deep CNN

The accuracy of a model that applies Deep CNN to predict the binary classification of engine flight capability is shown in the graph in Figure 11. The y-axis displays the model's accuracy, and the x-axis displays the number of epochs. At the conclusion of the training procedure, the model had an 87.99 % accuracy rate.

The graph presented in Figure 11 depicts the loss of a model that utilizes CNN to predict the binary classification of engine flight capability. The number of epochs is shown on the x-axis, and the y-axis shows the model loss. The model achieved a final loss of 0.2685.

The graph presented in Figure 12 depicts the actual and predicted values of a CNN model used to predict the binary classification of engine flight capability. Unfortunately, the graph suggests that the model could not accurately predict the value with a large discrepancy among the predicted and actual values. This graph highlights the limitations of the CNN model in this particular application and indicates the need for further development and exploration of other modelling techniques.

4.2 Remaining Useful Life Prediction

Predicting the RUL of aircraft engines using a variety of deep learning approaches [29], [30] was one of the goals of our research. The performance of each model was evaluated by plotting the mean squared error metrics and comparing the results. The plotted graphs offer a clear visual representation of the performance of each technique and allow us to determine which method is most effective for a given dataset.

1) LSTM+WOA

Figure 13's graph shows the mean squared error of an LSTM model combined with a WOA that was used to estimate the RUL left in an aviation engine. The number of epochs is shown on the x-axis, and the y-axis shows the mean squared error. The graph demonstrates that the model achieved a mean squared error of 1368.43 by the end of the training process. This graph provides

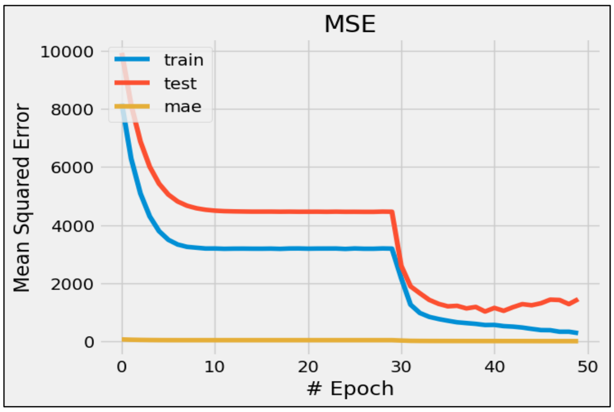

Figure 13. LSTM + WOA MSE graph.

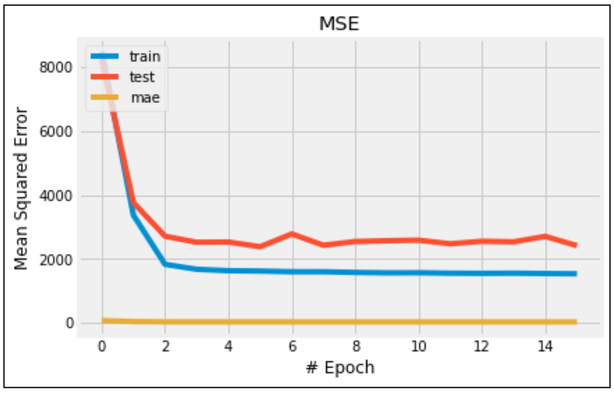

Figure 14. LSTM MSE graph

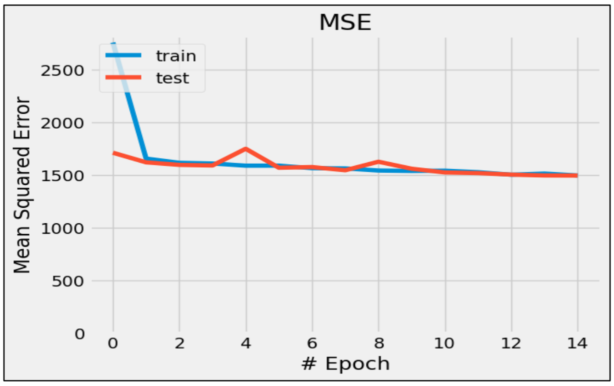

Figure 15. 1D - CNN MSE graph

a helpful instrument for evaluating the performance of the model and identifying areas for improvement, offering valuable insights into the potential of this technique for RUL prediction.

2) LSTM

The graph shown in Figure 14 shows the mean square error (MSE) of an LSTM model used to predict the RUL of aircraft engines. The y-axis shows the MSE while the x-axis shows the number of epochs. The graph shows that the model achieved an MSE of 2412.25 at the end of the training process.

3) 1D-CNN

The mean square error (MSE) of a 1D CNN model used to predict the residual useful life (RUL) of aircraft engines is shown in the graph in Figure 15. The number of epochs is shown on the x-axis, and the y-axis indicates the MSE. The graph shows

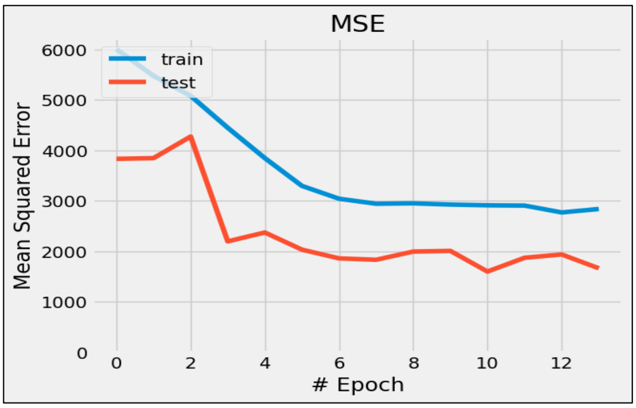

Figure 16. Deep CNN MSE graph.

Table 2. Result comparison for RUL prediction.

Metric	LSTM+WOA	LSTM	1D-CNN	DCNN
MSE	1368.43	2412.25	1492.18	1661.22

Table 3. Result comparison for binary classification.

Metric	LSTM+WOA	LSTM	1D-CNN	DCNN
Accuracy (in %)	98.32	95.53	90.05	87.99
Loss	0.032	0.1176	0.4001	0.2685

that the model had an MSE of 2353.1779 at the end of the training period.

4) Deep CNN

The graph shown in Figure 16 shows the mean square error (MSE) of a Deep CNN model used to predict the RUL of aircraft engines. The y-axis shows the MSE while the x-axis shows the number of epochs. The graph shows that the model achieved an MSE of 2232.49 at the end of the training process.

i) Remaining Useful Life Prediction

The Table 2 below summarizes the performance of various deep learning models using the mean squared error (MSE) metric. MSE measures the mean squared difference between predicted and actual values. A low MSE value indicates better performance, as it means that the predicted values are closer to the actual values.

ii) Binary Classification of Engine Flight Capability

The Table 3 below shows the values of accuracy and loss metrics used to compare the performance of various deep learning techniques.

The model's accuracy is defined by dividing the total number of predictions by the number of accurate ones. In other words, it is a measure of how well the model correctly classified the data. A higher accuracy score is considered better, as it indicates that the model is making fewer errors. On the other hand, the loss of the model is the difference between the predicted and actual values, calculated using a loss function. A lower loss score is considered better, as it indicates that the predicted values are closer to the actual values.

5. CONCLUSIONS

The LSTM model, 1D CNN model and Deep CNN model achieved a mean squared error (MSE) of 2412.25, 2353.18 and 2232.49, respectively at the end of training, indicating that the predictions were not very accurate. The LSTM + WOA model, on the other hand, achieved a significantly lower MSE of 1368 at the end of training, indicating that the predictions were more accurate compared to the LSTM model. In the context of predicting the binary classification of engine flight capability, the LSTM model attained a precision of 95.53 %, which is a good performance. The CNN model also attained accuracy of 94 %. However, when the same task was performed using LSTM hybridized with WOA, the accuracy increased significantly to 98.32 %. It should also be considered that the time taken by LSTM + WOA model is almost double the time taken by LSTM model. Therefore, the decision between the two models would depend on the particular use case and the accuracy/computability trade-off. The proposed research work has two objectives: to predict if an engine is capable of taking flight based on its Remaining Useful Life Parameter if available, and if not, to first predict the RUL by looking at its sensor measurement values and using regression principles to generate this value. Aircraft engine failure is a very serious issue given the sudden increase in the number of civilian and military aircraft that are currently in service. Numerous studies have gone into finding the reasons for faults in engines that contribute to engine failure. However, our project aims to predict the number of safe flights an engine can take, depending on how its sensors work currently. This serves as a useful precautionary study with significant impact to the aviation industry. With improved accuracy in this field, we can be guaranteed of safer flights, which can save both human lives and maintenance costs. The results suggest that the LSTM model hybridized with WOA performs better than the CNN model and the basic LSTM model for both classification and regression tasks related to predicting the performance of aircraft engines.

The limitations of the hybridized model in the real time data are Training Complexity, Non-Differentiability and Hyper Parameter Tuning.

The same model can be used to optimize the performance of Financial Time Series Prediction, Energy Consumption Forecasting, Healthcare Predictive Modelling, Smart Grid Optimization.

ACKNOWLEDGEMENT

Authors acknowledge the support from Ramaiah Institute of Technology, REVA University, Nagarjuna College of Engineering & Technology for the facilities provided to carry out the research.

REFERENCES

[1] A. Bala, I. Ismail, R. Ibrahim, S. M. Sait, D. Oliva, An improved grasshopper optimization algorithm based echo state network for predicting faults in airplane engines. IEEE Access, 8, pp. 159773-159789.
DOI: 10.1109/Access.2020.3020356

[2] A. P. Hermawan, D. S. Kim, J. M. Lee, Predictive maintenance of aircraft engine using deep learning technique, Int. Conf. on Information and Communication Technology Convergence (ICTC), Jeju, South Korea, 21-23 October 2020, pp. 1296-1298.
DOI: 10.1109/ICTC49870.2020.9289466

[3] F. A. Adryan, K. W. Sastra, Predictive Maintenance for Aircraft Engine Using Machine Learning: Trends and Challenges. AVIA, 3(1), pp. 37-44.
DOI: 10.47355/avia.v3i1.45

[4] R. Li, W. J. Verhagen, R. Curran, Toward a methodology of requirements definition for prognostics and health management system to support aircraft predictive maintenance. Aerospace

Science and Technology, 102, 10587.
DOI: 10.1016/j.ast.2020.105877

[5] K. Khan, M. Sohaib, A. Rashid, S. Ali, H. Akbar, A. Basit, T. Ahmad, Recent trends and challenges in predictive maintenance of aircraft's engine and hydraulic system. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 43, pp. 1-17. DOI: 10.1007/s40430-021-03121-2

[6] I. de Pater, A. Reijns, M. Mitici, Alarm-based predictive maintenance scheduling for aircraft engines with imperfect Remaining Useful Life prognostics. Reliability Engineering & System Safety, 221, 108341.
DOI: 10.1016/j.ress.2022.108341

[7] I. Stanton, K. Munir, A. Ikram, M. El Bakry, Predictive maintenance analytics and implementation for aircraft: Challenges and opportunities. Systems Engineering, Volume26, Issue2, March 2023, pp. 216-237
DOI: 10.1002/sys.21651

[8] W. J. Verhagen, L. W. De Boer, Predictive maintenance for aircraft components using proportional hazard models. Journal of Industrial Information Integration, 12, pp. 23-30.
DOI: 10.1016/j.jii.2018.04.004

[9] J. Lee, M. Mitici, Deep reinforcement learning for predictive aircraft maintenance using probabilistic Remaining-Useful-Life prognostics, Reliability Engineering & System Safety, Volume 230, 2023, 108908.
DOI: 10.1016/j.ress.2022.108908

[10] Ch. Chen, J. Shi, N. Lu, Zh. Hong Zhu, B. Jiang, Data-driven predictive maintenance strategy considering the uncertainty in remaining useful life prediction, Neurocomputing, Volume 494, 2022, pp. 79-88.
DOI: 10.1016/j.neucom.2022.04.055

[11] A. Fitra Azyus, S. K. Wijaya, M. Naved, Determining RUL Predictive Maintenance on Aircraft Engines Using GRU. Journal of Mechanical, Civil and Industrial Engineering, 3(3) 2022, pp. 79–84.
DOI: 10.32996/jmcie.2022.3.3.10

[12] C. Ordóñez, F. Sánchez Lasheras, J. Roca-Pardiñas, F. J. de Cos Juez, A hybrid ARIMA–SVM model for the study of the remaining useful life of aircraft engines, Journal of Computational and Applied Mathematics, Volume 346, 2019, pp. 184-191. DOI: 10.1016/j.cam.2018.07.008

[13] M. D. Dangut, I. K. Jennions, S. King, Z. Skaf, Application of deep reinforcement learning for extremely rare failure prediction in aircraft maintenance, Mechanical Systems and Signal Processing, 171, 2022, 108873.
DOI: 10.1016/j.ymssp.2022.108873

[14] M. D. Dangut, Z. Skaf, I. K. Jennions, An integrated machine learning model for aircraft components rare failure prognostics with log-based dataset, ISA transactions, 113 (2020), pp. 127-139. DOI: 10.1016/j.isatra.2020.05.001

[15] Ch. Che, H. Wang, Q. Fu, X. Ni, Combining multiple deep learning algorithms for prognostic and health management of aircraft. Aerospace Science and Technology, 94, 2019, 105423. DOI: 10.1016/j.ast.2019.105423

[16] M. Xiong, H. Wang, Q. Fu, Y. Xu, Digital twin–driven aero-engine intelligent predictive maintenance, The Int. Journal of Advanced Manufacturing Technology, 114(11-12), pp. 3751-3761.
DOI: 10.1007/s00170-021-06976-w

[17] P. J. Garcia Nieto, E. García-Gonzalo, F. Sánchez-Lasheras, F. J. de Cos Juez, Hybrid PSO–SVM–based method for forecasting of the remaining useful life for aircraft engines and Evaluation of its reliability, Reliability Engineering & System Safety, 138, 2015, pp. 219-231.
DOI: 10.1016/j.ress.2015.02.001

[18] V. Mathew, T. Toby, V. Singh, B. M. Rao, M. G. Kumar, Prediction of Remaining Useful Lifetime (RUL) of turbofan engine using machine learning, IEEE Int. Conf. on Circuits and Systems (ICCS), Thiruvananthapuram, India, 20-21 December 2017, pp. 306-311.
DOI: 10.1109/ICCS1.2017.8326010

[19] E. J. Amaya, A. J. Alvares, Prognostic of RUL based on echo state network optimized by artificial bee colony, Int. Journal of Prognostics and Health Management, 7(1) 2020, 12 pp.
DOI: 10.36001/ijphm.2016.v7i1.2359

[20] F. O. Heimes, Recurrent neural networks for remaining useful life estimation, Int. Conf. on Prognostics and Health Management, Denver, CO, USA, 6-9 October 2008, pp. 1-6.
DOI: 10.1109/PHM.2008.4711422

[21] X. Li, Q. Ding, J. Q. Sun, Remaining useful life estimation in prognostics using deep convolution neural networks, Reliability Engineering & System Safety, 172, 2018, pp. 1-11.
DOI: 10.1016/j.ress.2017.11.021

[22] G. S. Babu, P. Zhao, X. Li, Deep convolutional neural network based regression approach for estimation of remaining useful life, In: Navathe, S., Wu, W., Shekhar, S., Du, X., Wang, X., Xiong, H. (eds) Database Systems for Advanced Applications. DASFAA 2016. Lecture Notes in Computer Science(), vol 9642. Springer, Cham.
DOI: 10.1007/978-3-319-32025-0_14

[23] B. Zhang, Y. Li, Y. Bai, Y. Cao, Aeroengines Remaining Useful Life Prediction Based on Improved C-Loss ELM, IEEE Access, vol. 8, 2020, pp. 49752-49764.
DOI: 10.1109/access.2020.2979747

[24] R. Zhao, R. Yan, J. Wang, K. Mao, Learning to monitor machine health with convolutional bi-directional LSTM networks, Sensors, 17(2), 273.
DOI: 10.3390/s17020273

[25] Y. Wu, M. Yuan, S. Dong, L. Lin, Y. Liu, Remaining useful life estimation of engineered systems using vanilla LSTM neural networks, Neurocomputing, 275, pp. 167-179.
DOI: 10.1016/j.neucom.2017.05.063

[26] S. G., M. R. Mundada, S. Supreeth, B. Gardiner, Deep Learning-based Resource Prediction and Mutated Leader Algorithm Enabled Load Balancing in Fog Computing, Int. Journal of Computer Network and Information Security, vol. 15, no. 4. MECS Publisher, Aug. 08, 2023, pp. 84–95.
DOI: 10.5815/ijcnis.2023.04.08

[27] H. V. Ramachandra, Pundalik Chavan, S. Supreeth, H. C. Ramaprasad, K. Chatrapathy, G. Balaraju, S. Rohith, H. S. Mohan, Secured Wireless Network Based on a Novel Dual Integrated Neural Network Architecture, Journal of Electrical and Computer Engineering, vol. 2023, Sep. 28, 2023, pp. 1–11.
DOI: 10.1155/2023/9390660

[28] Pradeep Kumar D, Sowmya B J, A. Kanavalli, Amaresh T N, Anish S, Chinmay S Nadgir, Gagan A Nischal, Supreeth S, Shruthi G, Adaptive Hybridized Meta-Heuristic Algorithm for Subspace Clustering on High Dimensional Data, Journal of Advances in Information Technology, vol. 14, no. 5. Engineering and Technology Publishing, 2023, pp. 1103–1116.
DOI: 10.12720/jait.14.5.1103-1116

[29] Sh. Yamin Siddiqui, I. Ahmad, M. Adnan Khan, B. Shoaib Khan, M. Nadeem Ali, I. Naseer, K. Parveen, H. M. Usama, AIoT Enabled Traffic Congestion Control System Using Deep Neural Network, ICST Transactions on Scalable Information Systems, vol. 8, no. 33. European Alliance for Innovation n.o., p. 171170, Oct. 29, 2021.
DOI: 10.4108/eai.28-9-2021.171170

[30] N. Venkateswaran, S. P. Prabaharan, An Efficient Neuro Deep Learning Intrusion Detection System for Mobile Adhoc Networks, ICST Transactions on Scalable Information Systems, vol. 9, no. 6, 2022, p. 173781.
DOI: 10.4108/eai.4-4-2022.173781