Optimizations in Applied Machine Learning
Research Article | Volume 3 | Issue 3 | Oct 2023
Received: 2 Sep 2023 | Revised: 28 Sep 2023
Accepted: 8 Oct 2023 | Published Online: 15 Oct 2023

Residual Self-Attention-Based Temporal Deep Model for Predicting Aircraft Engine Failure within a Specific Cycle

Tong Zhou [1], Guojun Zhang [2,*], Yiqun Cai [3]

[1] Air China Cargo Co., Ltd., Beijing, 101318, CHINA,

[2] Corresponding, Quadrant international inc., San Diego, 92121, USA,
Email: ez900113@gmail.com

[3] University of Florida, Herbert Wertheim College, FL, 32608, USA

Abstract: Aircraft engines are complex, high-performance machines operating under extreme conditions, where reliability is critical for aviation safety. Early detection of engine faults is essential not only to ensure passenger safety but also to reduce operational costs and maintain efficient flight schedules. Traditional fault detection methods rely on rule-based diagnostics, setting predetermined thresholds for engine parameters such as temperature and pressure. However, these methods are limited in accuracy and often fail to capture the intricate degradation patterns of engine components, leading to late fault detection and frequent false alarms. Recent advancements in machine learning, particularly in deep learning, offer promising alternatives. Machine learning models can analyze large-scale time-series data and recognize complex patterns that human expertise might overlook. Among these, Long Short-Term Memory (LSTM) networks, combined with self-attention mechanisms, have shown potential in capturing temporal dependencies crucial for predictive maintenance. This study proposes a Residual Self-Attention-based LSTM model for predicting aircraft engine failures. By integrating residual connections with self-attention, the model enhances pattern recognition and interpretability, offering improved fault prediction accuracy over traditional models. The model was trained on aircraft engine sensor data, achieving high performance across various metrics, suggesting that this architecture holds significant promise for proactive aircraft maintenance applications.

Keywords: Component; Aircraft Engine Failure Prediction; Self-attention; Residual Connection.

1. Introduction

Aircraft engines are complex, high-performance machinery that operate under extreme conditions, making their reliability critical to aviation safety [1][2]. The prediction and early detection of engine faults are essential not only for ensuring passenger safety but also for minimizing operational costs and maintaining efficient scheduling. Faults in aircraft engines can lead to severe consequences, including in-flight engine shutdowns, which jeopardize passenger safety and entail significant financial losses. Therefore, the timely and accurate prediction of engine failure is paramount, and recent advancements in machine learning offer promising avenues for improved predictive models in this field.

In the past, traditional methods for aircraft engine fault detection relied primarily on rule-based diagnostics and condition monitoring systems due to their simplicity in many tasks [3][4][5]. These conventional techniques typically involve setting predetermined thresholds for various engine parameters, such as temperature, pressure, and vibration levels. When these values exceed safe thresholds, alerts are triggered, and maintenance actions are recommended. However, such rule-based systems often fall short in accuracy and flexibility. They tend to generate false alarms because they cannot fully capture the complexities of an engine's degradation process. Furthermore, these methods are reactive; they detect problems only after certain symptoms have manifested, which can be too late to prevent engine failure. Additionally, traditional methods rely heavily on expert knowledge [6][7][8], which may vary in consistency and quality, leading to limitations in the system's reliability. Given these constraints, the aviation industry requires a more robust solution that can proactively predict engine failures before critical faults occur.

Recent advances in machine learning (ML) have transformed predictive maintenance approaches by enabling models to analyze large volumes of time-series data and identify patterns that human experts might overlook [9][10][11]. Machine learning-based predictive models have demonstrated considerable advantages over traditional methods in terms of both accuracy and scalability [12][13][14]. For instance, Xiong et al. introduced a novel Multifunctional End-to-End Model for Optical Character Classification and Denoising that integrates denoising and recognition processes using a dual-output autoencoder, achieving significant gains in OCR accuracy and efficiency, especially in noisy and degraded image conditions [11]. ML techniques can handle complex relationships [15][16][17][18] among multiple engine parameters, providing a comprehensive analysis of the engine's health status. In particular, deep learning models have shown remarkable success in temporal sequence analysis [19][20][21], which is crucial for predicting engine performance over time. These models can automatically extract meaningful features from raw data, reducing the dependency on domain-specific knowledge and enhancing the predictive accuracy of the system.

Among ML techniques, recurrent neural networks (RNNs) [22][23][24], especially Long Short-Term Memory (LSTM) networks [25][26], have become popular for sequential data analysis due to their ability to retain long-term dependencies. LSTM networks are well-suited for analyzing time-series data, such as engine sensor readings, because they can remember information over extended sequences, which is essential for capturing the gradual degradation in engine performance. In recent studies, LSTM-based models have shown promising results in the field of predictive maintenance. These models are capable of learning from historical data and forecasting future trends, allowing them to predict engine failures within a specific operating cycle.

However, despite the success of LSTMs in predictive maintenance, there are still challenges when applying them to complex systems like aircraft engines. One of the main limitations is the lack of interpretability in LSTM models, making it difficult to understand which features contribute to predictions. Additionally, traditional LSTM models might struggle to capture intricate dependencies between different temporal features without further enhancement. To address these limitations, researchers have explored the integration of attention mechanisms with LSTM networks. The self-attention mechanism, in particular, has gained popularity in recent years as a powerful tool for enhancing the model's ability to focus on critical parts of the input sequence. By assigning different weights to various points in the input sequence, self-attention allows the model to identify and emphasize the most relevant features for prediction.

The proposed study aims to utilize a Residual Self-Attention-Based Temporal Deep Model shown in Figure 1 for predicting aircraft engine failure within a specific cycle. By incorporating residual connections with self-attention layers, this approach seeks to address the limitations of traditional

LSTM models in two significant ways. First, the residual connections enhance the model's capacity to capture deep and complex patterns without vanishing gradient issues, enabling it to retain relevant information over longer sequences. Second, the self-attention mechanism allows the model to focus selectively on important segments of the data, improving interpretability and making the model more robust in analyzing complex temporal dependencies among engine parameters. These enhancements are particularly valuable in a predictive maintenance context, where understanding which factors contribute most to the likelihood of engine failure can provide actionable insights for maintenance teams.

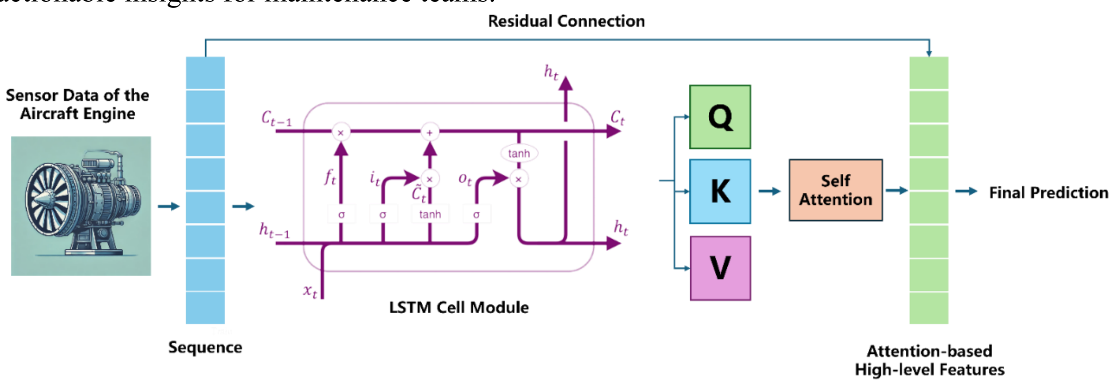

Figure 1. The workflow of the proposed residual self-attention-based temporal deep model.

2. **Literature Review**

A. Aircraft engine failure prediction based on machine learning

The field of aircraft engine prognostics is rapidly advancing, and the use of machine learning techniques to predict faults has become a focal point because of its ability to improve maintenance approaches and enhance system reliability. Recent studies have investigated a range of methods, showcasing the use of both conventional machine learning and sophisticated deep learning techniques. These approaches are valued for their superior performance in numerous applications.

Traditional machine learning techniques have been extensively employed in predictive maintenance scenarios. For instance, reference [27] discusses the classification of bearing defects using acoustic and vibrational signals that exhibit high noise levels and nonlinearity. The permutation entropy method, introduced in [28], is effectively used to analyze these complex time series data, followed by the implementation of the support vector machine (SVM) method for defect categorization. Additionally, reference [29] explores the successful use of SVM in regression tasks and examines the capability to estimate the remaining useful life (RUL) of aircraft engines based on unstructured and noisy data. Principal component analysis (PCA) [30][31][32] is applied to study the signals, employing a refined likelihood metric to compare systems under RUL evaluation with those in the training set. This methodology ensures that predictions for test set systems are based on their similarities to systems in the training set, allowing for precise prediction of duty cycle durations for each engine.

3. **Method**

A. Dataset preparation

To accurately predict aircraft engine failure within a specific cycle, we used an extensive dataset available on Kaggle. This dataset captures multiple operational cycles from a variety of aircraft engines, comprising thousands of individual records across numerous engines. Each record features sensor measurements and operational settings, creating a 24-dimensional feature set. The visualization of these sensor features is illustrated in Figure 2. The dataset has been pre-split into training and testing sets. For the training set, considering the use of time series models for predictive analysis, we segmented it into sequences with each sequence length set to 50, meaning 50 consecutive sequences are utilized for prediction. Before inputting data into the model, we normalized the entire dataset. This task is formulated as a binary classification problem, with labels designated as 0 (failure within a specific cycle) and 1 (no failure within a specific cycle).

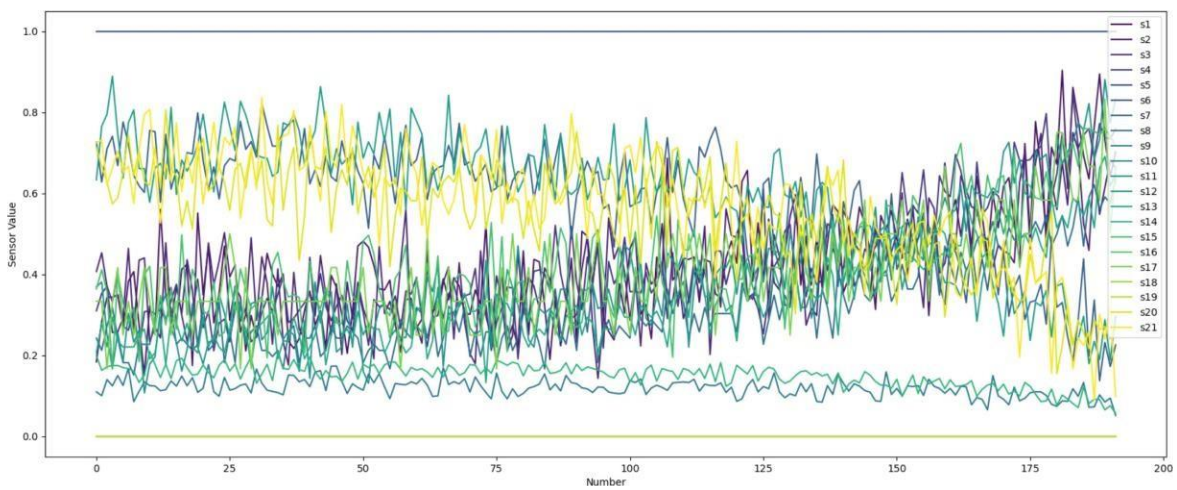

Figure 2. The visualization of sensor features in this dataset.

B. The residual self-attention-based LSTM model

1. Preliminaries of the LSTM

Long Short-Term Memory (LSTM) networks [33][34][35] are a type of recurrent neural network (RNN) capable of learning long-term dependencies in sequence prediction problems. Unlike standard feedforward neural networks, LSTMs have feedback connections that make them powerful for processing entire sequences of data. This makes LSTMs ideal for tasks such as time series prediction, natural language processing, and speech recognition.

The LSTM was introduced by Hochreiter & Schmidhuber in 1997 to specifically address the vanishing gradient problem that traditional RNNs face. The vanishing gradient problem occurs during backpropagation in deep networks when gradients are propagated back in time across many layers, causing them to shrink exponentially. This makes it difficult for the RNN to learn correlations between distant events. LSTMs solve this problem by incorporating memory cells that can maintain information in memory for long periods of time.

An LSTM unit includes three gates: the input gate, the output gate, and the forget gate. These gates determine whether to let new input in (input gate), delete the information (forget gate), or let it impact the output at the current timestep (output gate): 1) Input Gate: This gate decides the extent to which a new value flows into the cell. It involves a sigmoid activation layer that decides which values are allowed to update the memory state and a tanh layer that creates a vector of new candidate values that could be added to the state. 2) Forget Gate: It allows the cell to forget the previously stored information, depending on the new input and the previous output. This is crucial for the model to discard irrelevant information and prevent the neural network from becoming overwhelmed with too much information. 3) Output Gate: The output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit. The output gate takes the current input and the previous output into account, and decides which part of the current cell state will make it to the output.

LSTMs are particularly useful for learning sequences with varying time intervals and lengths. They have been successfully applied to predicting stock market trends, generating text, and even composing music. The ability to connect previous information to the current task (e.g., using previous video frames to predict the movement in a video) is what makes LSTMs superior to other models for sequence modeling tasks.

2. Preliminaries of the self-attention

Self-attention [36][37][38], a key component of the transformer architecture, is a mechanism that allows a model to weigh the significance of different parts of an input sequence independently of their position in the sequence. This approach has revolutionized natural language processing (NLP) and other sequence-based tasks by providing a flexible way of handling input data. The core idea behind self-attention is to compute the relevance of all parts of the input to each part of the output. In practice, this means that each output element, such as a word in a sentence, is expressed as a weighted sum of all input elements. This weighting determines how much attention or importance is given to each input element when computing a particular output.

Self-attention is implemented using three vectors for each input element: Query, Key, and Value. These vectors are derived by transforming the input elements through learned linear transformations. For a given element, the Query vector is used to compute a score against every Key vector from the other elements in the sequence. These scores determine the weights for how much each element's Value vector should be considered for the output. The scores are typically normalized using a softmax function to ensure they add up to one, representing probabilities.

This mechanism allows the model to focus on the most relevant parts of the input data, which is particularly useful in tasks like machine translation, where the relevance of input words can vary significantly depending on the context. By computing the attention dynamically for each pair of input and output, self-attention models can adaptively highlight or downplay features as needed, without relying on the rigid structure of recurrent layers.

Moreover, self-attention facilitates parallel processing of data, significantly speeding up training, as it does not require sequential processing like RNNs. This efficiency, combined with its ability to manage long-range dependencies in data, makes self-attention a powerful tool in building advanced neural network architectures.

3. Preliminaries of the residual connection

Residual connections [39][40], also known as skip connections, are a network architecture innovation introduced to address the problem of training very deep neural networks. As networks increase in depth, training them becomes challenging due to issues like vanishing gradients, where the gradients become too small to make significant updates, leading to stagnant training processes. Residual connections help mitigate this by allowing gradients to flow directly through the network's architecture via shortcuts past one or more layers.

Introduced by He et al. in their seminal paper on ResNet [41], residual connections revolutionized deep learning by enabling the development of networks that are significantly deeper than those that were previously feasible. The key idea is simple yet powerful: instead of trying to learn an underlying mapping directly, a layer in a network with a residual connection learns the difference (or residual) between the input and the output of the layer. The mathematical representation is $F(x)+x$, where x is the input to the layer, and $F(x)$ is the output of the transformation applied by the layer. This output is then added back to the original input.

This setup forms a shortcut or a direct path for the backward pass during training, ensuring that the gradient can be propagated directly back through the network without diminishing in strength, effectively addressing the vanishing gradient problem. The outputs of layers are thus the sum of their inputs and the residuals, meaning the network only needs to learn the adjustments rather than the entire transformation, making learning easier and more efficient.

4. The architecture of the proposed model

In this study, we propose a neural network architecture designed to enhance fault prediction capabilities in aircraft engine prognostics. Architecture integrates several layers aimed at effectively capturing and processing temporal relationships inherent in sequential data. Initially, the model defines input and output dimensions tailored to handle one output label for each sequence with a predetermined number of features per sequence.

Central to our architecture is a LSTM layer, which addresses the vanishing gradient problem typically associated with standard recurrent neural networks. This layer is configured to return sequences, facilitating a deeper temporal analysis across the input data. To mitigate the risk of overfitting, a dropout layer with a rate of 20% follows the LSTM layer, randomly omitting a subset of features during training. Subsequently, an attention mechanism is employed, focusing the model's capacity to weigh different parts of the sequence based on their relevance to the task at hand. This custom attention layer applies a softmax function [42] to the tanh-activated [43] product of the inputs and a learned weight matrix, producing a context vector that summarily represents the most salient features in the sequence. To further enhance the model’s ability to leverage both learned features and original input information, a residual connection is introduced. This connection adds the output from the attention layer to a transformed version of the initial sequence data, averaged over all time steps, thereby enabling the integration of deep contextual information with less-transformed inputs. The architecture concludes with an output layer featuring a dense network equipped with a sigmoid activation function, which is suitable for binary classification tasks. This setup ensures that the model delivers a probability distribution across possible outcomes, facilitating its application in scenarios requiring binary decisions.

C. Implementation details

The model is developed using the TensorFlow framework, configured to train about 30 epochs with a batch size of 200. The Adam optimizer is employed to enhance the training process by efficiently adjusting the weights based on the gradients. For evaluation, the model utilizes accuracy as the primary metric, which assesses the percentage of correctly predicted instances against the total predictions made, providing a clear measure of the model's performance.

4. **Results and Discussion**

A. The performance of the proposed residual self-attention-based LSTM model

The comprehensive evaluation of three distinct machine learning models shown in Table 1, Figure 3, Figure 4 and Figure 5, namely the proposed Residual Self-Attention-based LSTM model, the standard LSTM model, and the traditional RNN model, on a testing dataset, unveils intriguing insights into their performance capabilities across multiple metrics including accuracy, precision, recall, and F1-score. This detailed analysis delves into not only the numerical performance metrics but also the training curves and prediction samples, thereby offering a holistic view of each model's strengths and weaknesses.

Starting with the Residual Self-Attention-based LSTM model, it achieves a notable accuracy of 90.8%, which is the highest among the three models. This model incorporates a self-attention mechanism that allows it to prioritize information from more relevant parts of the data sequence, enhancing its ability to understand complex patterns. It achieves a precision of 87.5% and a recall of 81.3%, culminating in an F1-score of 84.3%. These metrics indicate that it not only makes correct predictions reliably but also maintains a balanced approach between precision and recall, effectively managing the trade-offs between these two metrics. The model's training curves reflect a consistent learning process, as evidenced by the accuracy and loss plots. The accuracy stabilizes around 90% early in training, demonstrating rapid learning and convergence. The loss curve shows a steep decline initially and flattens out, which suggests that the model quickly reduces error rates and then fine-tunes its parameters for optimal performance.

In contrast, the LSTM model, while simpler than its self-attention counterpart, also shows robust performance with an accuracy of 88.5%. It excels particularly in precision at 90.8%, which is the highest among the three models, suggesting that when it predicts an instance as positive, it is very likely to be correct. However, its recall of 64.9% is notably lower, which indicates some shortcomings in identifying all relevant instances. The F1-score of 75.7% reflects these dynamics, pointing to a model that is conservative in its predictions, prioritizing certainty over coverage. The training curves for the LSTM model reveal a slightly less stable learning process compared to the Residual LSTM. The accuracy curve oscillates more noticeably, which may suggest overfitting on the training data or sensitivity to the training batch composition. The loss curve follows a similar trajectory to the Residual LSTM but with less smoothness, further supporting the inference of potential overfitting issues.

The basic RNN model, despite being the simplest model architecture among the three, competes closely with an accuracy of 88.8%. Its precision and recall are reasonably balanced at 87.2% and 67%, respectively, leading to an F1-score of 75.8%. This model's performance is commendable given its architectural simplicity, and it underscores the potential effectiveness of RNNs in tasks that do not require long-term dependency recognition or complex pattern understanding. The training curves for the RNN model are the most stable among the three models, with both accuracy and loss exhibiting less fluctuation. This could indicate a better generalization to the validation data, although the ultimate performance ceiling is lower than the models with more sophisticated architectures.

Moreover, the prediction sample plots provide visual insights into how each model performs with actual test data. These plots show the predicted versus actual values over a sample of test data points. For the Residual Self-Attention-based LSTM model, the predictions closely follow the actual values, demonstrating the model's effective learning and prediction capabilities. In comparison, the LSTM and RNN models show slightly less alignment with the actual values, particularly in capturing the sharper peaks and troughs in the data, which may correspond to more nuanced aspects of the data that the basic RNN fails to capture entirely.

This extensive analysis indicates that while more complex models like the Residual Self-Attention-based LSTM offer significant advantages in handling datasets where the understanding of context and focus within sequences is crucial, simpler models like the LSTM and RNN can also achieve commendable results depending on the specific requirements of the task. Therefore, the choice of model should be guided by the specific nuances of the dataset and task requirements, balancing the need for accuracy, computational efficiency, and ease of training.

Table 1. The performance of different approaches in the testing dataset.

Model Name	Accuracy	Precision	Recall	F1-score
Residual self-attention-based LSTM model	0.908	0.875	0.813	0.843
LSTM model	0.885	0.908	0.649	0.757
RNN model	0.888	0.872	0.670	0.758

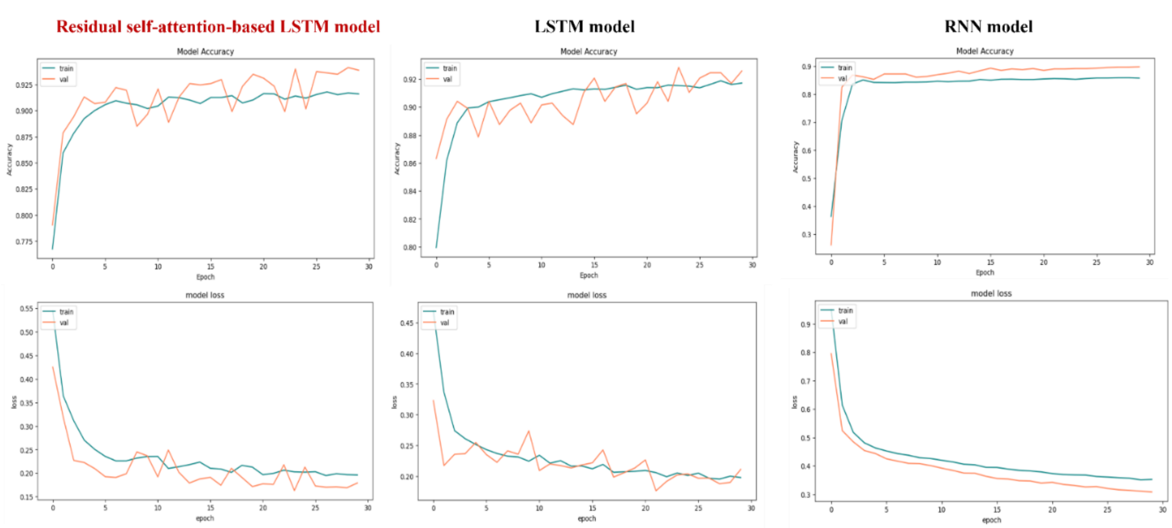

Figure 3. The training curves of different models.

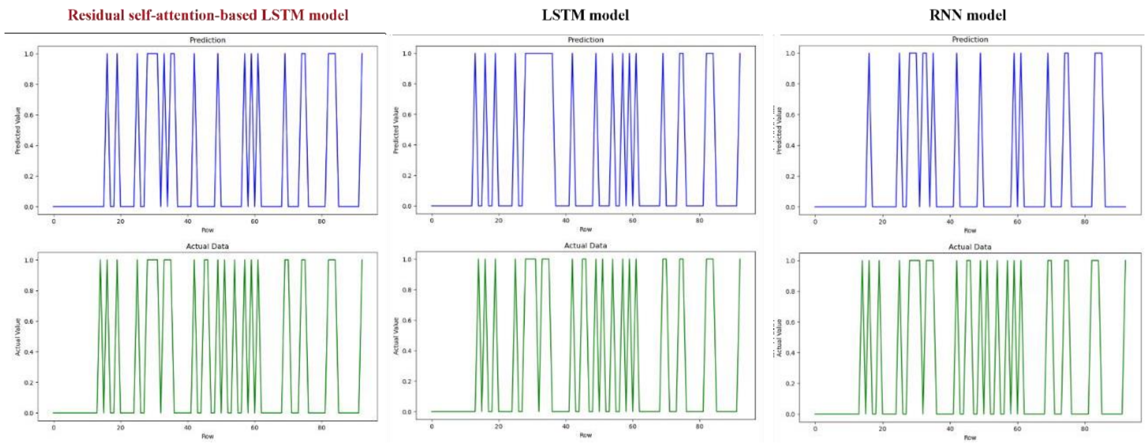

Figure 4. The prediction samples on the testing dataset.

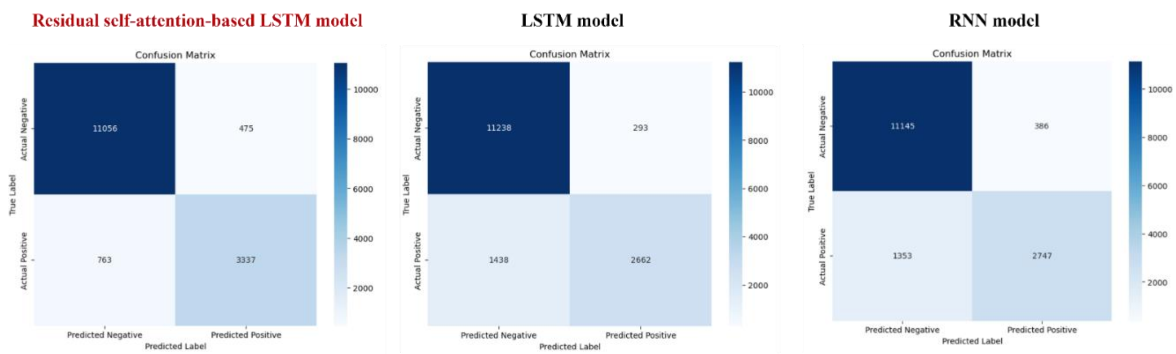

Figure 5. The confusion matrices of different models.

A. The influence of sequence number on the model performance

The bar chart illustrates the impact of different sequence lengths (50, 30, and 10) on the performance metrics of the Residual Self-Attention-based LSTM model. This analysis is crucial in understanding how the sequence length in time-series or sequence data affects model performance in terms of accuracy, precision, recall, and F1-score.

From the chart shown in Figure 6, it is evident that as the sequence length increases, there is a notable improvement in all the performance metrics. Specifically, the model with a sequence number of 50 showcases the highest scores across all metrics, indicating that a longer sequence length provides the model with more context and a better understanding of the dependencies in the data. This model achieves the highest precision and F1-score, which are critical for models where the cost of false positives and false negatives is high. Precision is particularly high, suggesting that the model is very reliable when it predicts positive classes, making it valuable in applications where precision is more critical than recall.

For sequence number 30, there is a slight decrease in all metrics compared to sequence number 50. However, the performance is still robust, suggesting that the model effectively captures and utilizes the temporal information in the sequences but might be missing some nuances that longer sequences capture. This could be a sweet spot for applications that require a balance between computational efficiency and model performance.

The model with sequence number 10 shows the lowest scores among the three configurations. This substantial drop, especially in recall and F1-score, indicates that shorter sequences provide insufficient context for the model to make accurate predictions. The limited data points in each sequence may lead to overfitting on less significant features, reducing the model's ability to generalize well on unseen data. This configuration might only be suitable for very specific applications where the computational cost is a critical factor, and the sequences inherently contain less temporal dependency.

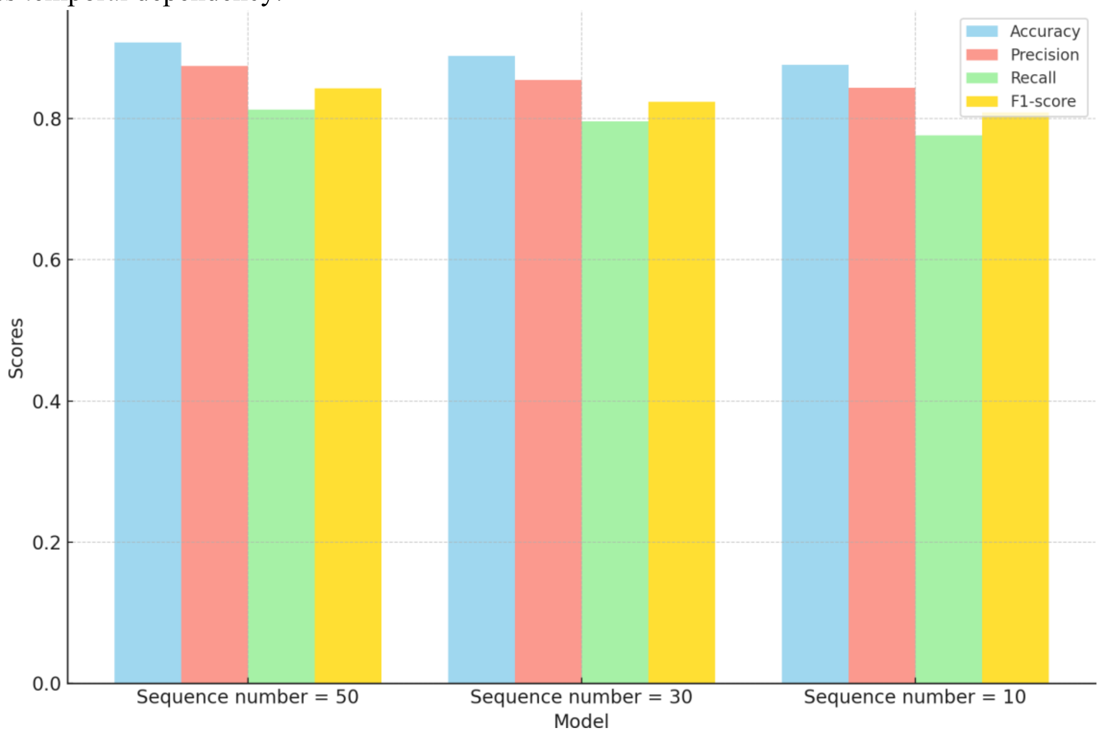

Figure 6. The influence of the sequence number on the model performance.

B. Ablation study

Table 2 presents an ablation study that compares three different models: the Residual Self-Attention-based LSTM model, the Self-Attention-based LSTM model, and a standard LSTM model. Each model's performance is evaluated based on four key metrics: accuracy, precision, recall, and F1-score.

Among the three models, the Residual Self-Attention-based LSTM model achieves the highest scores across most metrics, with an accuracy of 90.8%, precision of 87.5%, recall of 81.3%, and an F1-score of 84.3%. These results suggest that adding both self-attention and residual connections enhances the model’s ability to capture complex patterns in the data, resulting in better overall performance. The high recall of this model indicates it is particularly effective at identifying positive cases, which can be critical in applications where missing positive instances carries a high cost.

The Self-Attention-based LSTM model, which uses self-attention but lacks residual connections, shows slightly lower results across most metrics, with an accuracy of 89.9%, precision of 87.3%, recall of 80.6%, and an F1-score of 83.8%. While it still performs well, the absence of residual connections may reduce its effectiveness in capturing and retaining important information over long sequences. This model maintains a good balance between precision and recall, but it doesn't reach the top performance seen in the residual-enhanced version.

The standard LSTM model, without self-attention or residual connections, has the lowest scores overall, with an accuracy of 88.5%, precision of 90.8%, recall of 64.9%, and an F1-score of 75.7%. Although it achieves the highest precision, its recall is notably lower, indicating it struggles to identify as many positive cases. This model may perform well when precise positive predictions are required but could miss many relevant cases, which makes it less ideal for tasks where comprehensive detection is needed.

Table 2. The ablation study.

Model Name	Accuracy	Precision	Recall	F1-score
Residual self-attention-based LSTM model	0.908	0.875	0.813	0.843
self-attention-based LSTM model	0.899	0.873	0.806	0.838
LSTM model	0.885	0.908	0.649	0.757

C. Discussion

While this Residual Self-Attention-based LSTM approach is effective in capturing complex temporal dependencies and prioritizing critical information in sequential data, it also has several limitations.

First, this model architecture is computationally intensive. The combination of LSTM cells with self-attention layers, as well as the residual connections, requires substantial processing power and memory. LSTM networks are already known for their high computational cost due to the iterative processing of sequences, and the addition of self-attention layers exacerbates this, as it involves matrix operations that scale with the length of the sequence. This results in slower training and inference times, which may not be suitable for real-time applications or large datasets without access to advanced computational resources.

Second, this model's complexity may lead to challenges in interpretability. The residual connections and self-attention mechanisms, while useful for enhancing the model's performance, create a more opaque structure. Understanding why the model makes certain predictions becomes difficult, especially as the attention weights interact with the outputs of multiple LSTM states. This lack of transparency can be a drawback in fields like healthcare or finance, where model interpretability is essential for trust and regulatory compliance. In addition, tuning the various hyperparameters for both the LSTM and attention layers requires expertise and can be time-consuming, which may pose a barrier for some users.

5. **Conclusion**

The Residual Self-Attention-based LSTM model demonstrated superior performance in predicting aircraft engine faults compared to traditional LSTM and RNN models. By incorporating self-attention and residual connections, this model efficiently captures essential temporal dependencies, providing a more accurate and interpretable framework for predictive maintenance. The model achieved high precision and recall, highlighting its ability to reliably detect potential failures before they become critical, thus contributing to both safety and operational efficiency in aviation. However, while effective, the model also comes with challenges, including high computational demands and complexity in interpretation. These factors may limit its applicability in real-time systems without sufficient processing power. Nevertheless, the model's performance suggests it could be a valuable asset in scheduled predictive maintenance, helping maintenance teams make more informed decisions and reduce unexpected failures, ultimately promoting a safer and more efficient aviation industry.

Funding

This research has been partly funded by the National Natural Science Foundation China (NSFC) through award No.51971139. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSFC.

Author Contributions

Tong Zhou conceptualized the study, analyzed the data, and drafted the manuscript. Guojun Zhang supervised the study and provided critical revisions. Yiqun Cai contributed to the data analysis and manuscript revisions. All authors read and approved the final manuscript.

Institutional Reviewer Board Statement

Not applicable

Informed Consent Statement

Not applicable

Data Availability Statement

The data supporting the findings of this study are available from the corresponding author upon request.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] Kerrebrock, J. L. (1992). Aircraft engines and gas turbines. MIT press.

[2] Pollock, T. M. (2016). Alloy design for aircraft engines. Nature materials, 15(8), 809-815.

[3] Wenjun, D., Fatahizadeh, M., Touchaei, H. G., Moayedi, H., & Foong, L. K. (2023). Application of six neural network-based solutions on bearing capacity of shallow footing on double-layer soils. Steel and Composite Structures, 49(2), 231-244.

[4] Dai, W. (2023). Design of traffic improvement plan for line 1 Baijiahu station of Nanjing metro. Innovations in Applied Engineering and Technology.

[5] Ren, P., Zhao, Z., & Yang, Q. (2023). Exploring the Path of Transformation and Development for Study Abroad Consultancy Firms in China. Journal of Computational Methods in Engineering Applications, 1-12.

[6] Zhao, Z., Ren, P., & Yang, Q. (2023). Student Self-Management, Academic Achievement: Exploring the Mediating Role of Self-Efficacy and the Moderating Influence of Gender—Insights From a Survey Conducted in 3 Universities in America. Journal of Integrated Social Sciences and Humanities, 1-12.

[7] Dai, W. (2022). Evaluation and improvement of carrying capacity of a traffic system. Innovations in Applied Engineering and Technology, 1-9.

[8] Li, C., & Tang, Y. (2023). The Factors of Brand Reputation in Chinese Luxury Fashion Brands. Journal of Integrated Social Sciences and Humanities, 1-14.

[9] Dai, W. (2021). Safety evaluation of traffic system with historical data based on Markov process and deep-reinforcement learning. Journal of Computational Methods in Engineering Applications, 1-14.

[10] Xiong, S., Zhang, H., Wang, M., & Zhou, N. (2022). Distributed Data Parallel Acceleration-Based Generative Adversarial Network for Fingerprint Generation. Innovations in Applied Engineering and Technology, 1-12.

[11] Xiong, S., Chen, X., & Zhang, H. (2023). Deep Learning-Based Multifunctional End-to-End Model for Optical Character Classification and Denoising. Journal of Computational Methods in Engineering Applications, 1-13.

[12] Xiong, S., Zhang, H., & Wang, M. (2022). Ensemble Model of Attention Mechanism-Based DCGAN and Autoencoder for Noised OCR Classification. Journal of Electronic & Information Systems, 4(1), 33-41.

[13] Yu, L., Li, J., Cheng, S., Xiong, S., & Shen, H. (2013). Secure continuous aggregation in wireless sensor networks. IEEE Transactions on Parallel and Distributed Systems, 25(3), 762-774.

[14] Xiong, S., Yu, L., Shen, H., Wang, C., & Lu, W. (2012, March). Efficient algorithms for sensor deployment and routing in sensor networks for network-structured environment monitoring. In 2012 Proceedings IEEE INFOCOM (pp. 1008-1016). IEEE.

[15] Feng, Z., Xiong, S., Cao, D., Deng, X., Wang, X., Yang, Y., ... & Wu, G. (2015, March). Hrs: A hybrid framework for malware detection. In Proceedings of the 2015 ACM International Workshop on International Workshop on Security and Privacy Analytics (pp. 19-26).

[16] Xiong, S., Li, J., Li, M., Wang, J., & Liu, Y. (2011, April). Multiple task scheduling for low-duty-cycled wireless sensor networks. In 2011 Proceedings IEEE INFOCOM (pp. 1323-1331). IEEE.

[17] Yu, L., Li, J., Cheng, S., & Xiong, S. (2011, April). Secure continuous aggregation via sampling-based verification in wireless sensor networks. In 2011 Proceedings IEEE INFOCOM (pp. 1763-1771). IEEE.

[18] Chen, X., & Zhang, H. (2023). Performance Enhancement of AlGaN-based Deep Ultraviolet Light-emitting Diodes with AlxGa1-xN Linear Descending Layers. Innovations in Applied Engineering and Technology, 1-10.

[19] Zhou, Z., Wu, J., Cao, Z., She, Z., Ma, J., & Zu, X. (2021, September). On-Demand Trajectory Prediction Based on Adaptive Interaction Car Following Model with Decreasing Tolerance. In 2021 International Conference on Computers and Automation (CompAuto) (pp. 67-72). IEEE.

[20] Wang, H., Li, J., & Xiong, S. (2008). Efficient join algorithms for distributed information integration based on XML. International Journal of Business Process Integration and Management, 3(4), 271-281.

[21] Xiong, S., & Li, J. (2009, April). Optimizing many-to-many data aggregation in wireless sensor networks. In Asia-Pacific Web Conference (pp. 550-555). Berlin, Heidelberg: Springer Berlin Heidelberg.

[22] Medsker, L. R., & Jain, L. (2001). Recurrent neural networks. Design and Applications, 5(64-67), 2.

[23] Grossberg, S. (2013). Recurrent neural networks. Scholarpedia, 8(2), 1888.

[24] Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent advances in recurrent neural networks. arXiv preprint arXiv:1801.01078.

[25] Graves, A., & Graves, A. (2012). Long short-term memory. Supervised sequence labelling with recurrent neural networks, 37-45.

[26] Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. Artificial Intelligence Review, 53(8), 5929-5955.

[27] Vakharia, V., Gupta, V. K., & Kankar, P. K. (2015). Ball bearing fault diagnosis using supervised and unsupervised machine learning methods. Int J Acoust Vib, 20(4), 244-250.

[28] Bandt, C., & Pompe, B. (2002). Permutation entropy: a natural complexity measure for time series. Physical review letters, 88(17), 174102.

[29] Chen, Z., Cao, S., & Mao, Z. (2017). Remaining useful life estimation of aircraft engines using a modified similarity and supporting vector machine (SVM) approach. Energies, 11(1), 28.

[30] Abdi, H., & Williams, L. J. (2010). Principal component analysis. Wiley interdisciplinary reviews: computational statistics, 2(4), 433-459.

[31] Jolliffe, I. (2005). Principal component analysis. Encyclopedia of statistics in behavioral science.

[32] Shlens, J. (2014). A tutorial on principal component analysis. arXiv preprint arXiv:1404.1100.

[33] Laghrissi, F., Douzi, S., Douzi, K., & Hssina, B. (2021). Intrusion detection systems using long short-term memory (LSTM). Journal of Big Data, 8(1), 65.

[34] Kratzert, F., Klotz, D., Brenner, C., Schulz, K., & Herrnegger, M. (2018). Rainfall–runoff modelling using long short-term memory (LSTM) networks. Hydrology and Earth System Sciences, 22(11), 6005-6022.

[35] Huang, R., Wei, C., Wang, B., Yang, J., Xu, X., Wu, S., & Huang, S. (2022). Well performance prediction based on Long Short-Term Memory (LSTM) neural network. Journal of Petroleum Science and Engineering, 208, 109686.

[36] Shaw, P., Uszkoreit, J., & Vaswani, A. (2018). Self-attention with relative position representations. arXiv preprint arXiv:1803.02155.

[37] Zhao, H., Jia, J., & Koltun, V. (2020). Exploring self-attention for image recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10076-10085).

[38] Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019, May). Self-attention generative adversarial networks. In International conference on machine learning (pp. 7354-7363). PMLR.

[39] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017, February). Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI conference on artificial intelligence (Vol. 31, No. 1).

[40] Xie, S., Zhang, H., Guo, J., Tan, X., Bian, J., Awadalla, H. H., ... & Yan, R. (2023). Residual: Transformer with dual residual connections. arXiv preprint arXiv:2304.14802.

[41] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[42] Banerjee, K., Gupta, R. R., Vyas, K., & Mishra, B. (2020). Exploring alternatives to softmax function. arXiv preprint arXiv:2011.11538.

[43] Lau, M. M., & Lim, K. H. (2018, December). Review of adaptive activation function in deep neural network. In 2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES) (pp. 686-690). IEEE.