Enhanced Anomaly Detection in Aero-Engines using Convolutional Transformers

M.R. Babaei. A.D. Fentaye, K. Kyprianidis

School of Business, Society and Engineering (EST), Mälardalen University,72220 Västerås, Sweden

(e-mail: mohammad.reza.babaei@mdu.se, amare.desalegn.fentaye@mdu.se, konstantinos.kyprianidis@mdu.se)

Abstract: Gas turbines are vital in power generation and propulsion systems. However, these engines are exposed to complex and variable operating conditions, which makes early and accurate fault detection essential for predictive maintenance and minimizing unplanned downtime. This paper proposes a novel approach that combines convolutional neural networks (CNNs) with transformer architectures to address these challenges. The proposed Convolutional transformer model aims to enhance the accuracy and robustness of turbofan fault classification by integrating the feature extraction capabilities of CNNs with the contextual learning strengths of transformers. Through rigorous experiments, we seek to demonstrate our approach's performance in classification accuracy and generalization across different operating conditions. We utilize a comprehensive synthetic dataset, C- MAPSS, derived from multiple aircraft engine units as the benchmark for this study. The results for the proposed model show an accuracy of 99.6% on the test dataset. The outcome has the potential to be extended and fine-tuned for different types of gas turbines for diverse applications.

Keywords: fault classification, gas turbine engines, attention mechanism, convolutional neural network, machine learning

1. INTRODUCTION

The reliable operation of turbofan gas turbines is crucial to ensuring the safety and efficiency of modern aviation. As these engines are exposed to complex and variable operating conditions, early and accurate fault detection is essential for predictive maintenance and minimizing unplanned downtime. Fault classification plays a pivotal role in identifying and rectifying potential issues before they become significant failures.

The classification of faults in turbofan gas turbines is confronted with multiple significant challenges (Fentaye et al. (2019); X. Yang et al. (2023); Z. Yang et al. (2013)). Firstly, these engines are fitted with numerous sensors that monitor a range of parameters such as temperature, pressure, and vibration, leading to the generation of high-dimensional data streams that are challenging to process and analyze effectively. Secondly, the operational behavior of turbofan engines is characterized by complex temporal patterns, as faults often develop progressively over time, necessitating the capture of these temporal dependencies for accurate classification. Additionally, faults can progress at varying rates depending on operating conditions, complicating the development of models that can generalize across different scenarios. Moreover, many datasets exhibit an imbalance in fault classes, where some fault types are significantly more prevalent than others, potentially leading to biased classification models. Finally, for practical implementation, fault classification models must operate in real-time or near-real-time, necessitating the use of efficient algorithms capable of managing large volumes of streaming data.

In recent years, various machine learning (ML) and deep learning approaches have been explored for turbofan gas turbine data exploration and fault classification. A study by Xie et al. (2023) shows that feature extraction is a vital step in the ML pipeline, involving the transformation of raw sensor readings into informative and discriminative features that can improve the performance of classification models. Barrera et al. (2022) combines gained information from clustering with an auto-encoder for anomaly detection. Traditional techniques like support vector machines (SVM) and random forests have been used with reasonable success. For instance, Zhou et al. (2015) applied SVMs to classify different fault conditions with reasonable accuracy. Losi et al. (2022) were utilized random forest (RF) to predict gas turbine trips based on the snapshot matrix of records. However, these methods often struggle to oversee the high dimensionality and complex temporal dependencies inherent in turbine data.

The advent of deep learning has brought significant advancements in fault classification. CNNs have been highly effective due to their ability to extract hierarchical features from raw sensor data automatically Sateesh Babu et al. (2016) for remaining useful life (RUL) estimation. Long short-term memory (LSTM) networks have also been employed in research by Cao et al. (2021) to capture temporal dependencies, further improving classification performance. A study by Arpit et al. (2019) suggests a variant of LSTM to overcome well-known limitations like the vanishing gradient discussed by Pascanu et al. (2012). Fentaye et al. (2021) implements a Deep CNN to detect and isolate multiple gas path faults with over 96% accuracy. Despite these advances,

existing models still face challenges balancing local feature extraction and capturing global dependencies within the data.

This paper proposes a novel approach that combines CNNs with transformer architectures to address these challenges. Transformers, originally introduced by Vaswani et al. (2017) for natural language processing, have shown exceptional performance in capturing long-range dependencies and contextual information. By integrating the feature extraction capabilities of CNNs with the contextual learning strengths of transformers, our proposed convolutional transformer model aims to enhance the accuracy and robustness of turbofan gas turbine fault classification.
We will conduct a comparative analysis of the Convolutional transformer model against other state-of-the-art models, including a feed-forward transformer and standalone CNN models. Through rigorous experiments, we seek to demonstrate our approach's superiority in classification accuracy and generalization across different operating conditions. The proposed model should be able to explore and learn imbalanced dataset with complex dynamics effectively. In addition, address common issues with recurrent neural networks and provide the possibility to extend into deep neural networks (DNNs). This research contributes to advancing fault classification methodologies and has significant implications for implementing predictive maintenance in real-world aviation scenarios.

2. METHODOLOGY

2.1 Data collection

This study uses a comprehensive synthetic dataset published by Arias Chao et al. (2021) from multiple aircraft engine units, the NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS). C-MAPSS is a turbofan engine degradation dataset which is a widely used dataset in the field of predictive maintenance and prognostics.

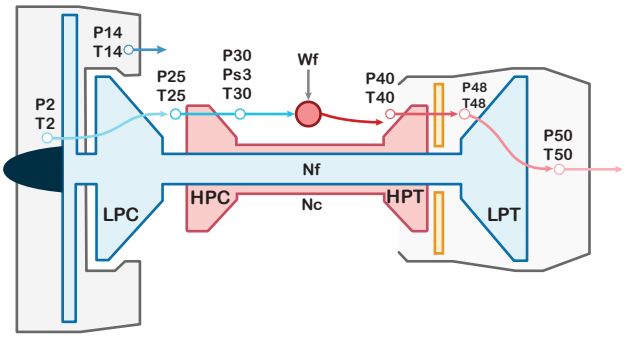

Fig. 1. A schematic of a turbofan engine gas turbine

It was developed to simulate realistic engine degradation scenarios under various operational conditions, providing researchers and practitioners with a valuable resource for developing and benchmarking fault detection and remaining useful life (RUL) prediction algorithms. The dataset comprises multiple sets of time-series data, capturing different degradation patterns across several turbofan engines. The C-MAPSS dataset is instrumental for advancing ML and statistical methods in predictive maintenance, allowing for the creation and validation of models that can predict equipment failures, optimize maintenance schedules, and ultimately enhance the reliability and safety of aerospace systems. Nevertheless, there has been relatively limited research focused on fault diagnostics and classification in gas turbines, indicating a significant opportunity for further exploration of advanced ML methods in anomaly detection within this context. The variables in the dataset are categorized into scenario descriptors ω, health parameters θ, measurements $\hat{x}_s$, and virtual sensors $\hat{x}_v$. Figure 1 shows a schematic of turbofan engine gas turbine.
This dataset consists of 9 units (2, 5, 10, 11, 14, 15, 16, 18, and 20). The units 2, 5, and 10 are subject to degradation of the efficiency of the high-pressure turbine (HPT) (Fault 1). The units 11, 14, 15, 16, 18, and 20 are subject to the flow and efficiency of low-pressure turbine (LPT) in addition to HPT efficiency degradation (Fault 2). Figure 2 shows the health parameters corresponding to the faults occurring within the dataset.

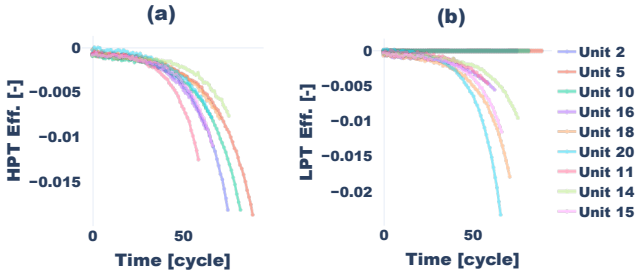

Fig. 2. Health parameters for LPT and HPT during the cycles

To eliminate the effects of ambient variables on the measure, the data goes through a correction process based on Volponi (2020). For example, the ambient temperature affects the system temperatures, pressure, and, consequently, shaft speed and fuel flow. Although the effect of the inlet temperature on the pressure is negligible, according to (1-3), the corrected values of temperatures, fuel flow, and shaft speed replaced the raw values to analyze the system's state during the operation.

$$\theta = \frac{T_{amb}}{T_{ref}} \quad , \quad T_{ref} = 288.15\ K \tag{1}$$

$$T_i^{corr} = T_i/\theta \tag{2}$$

$$W_f^{corr} = \frac{W_f}{\sqrt{\theta}} \quad , \quad N_i^{corr} = \frac{N_i}{\sqrt{\theta}} \tag{3}$$

Figure 3 demonstrates the distribution of total pressure at LPT outlet pressure (P50) during cycles for different units in the dataset.

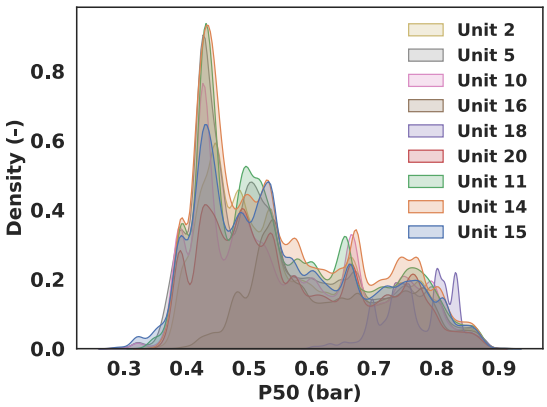

Fig. 3. LPT outlet pressure distribution for the engine units in the dataset

2.2 Attention mechanism

The transformer model, initially introduced by Vaswani et al. (2017), has become a cornerstone in natural language processing and sequence modeling tasks. The standard transformer consists of an encoder and a decoder, comprising stacked self-attention and feedforward (FF) neural network layers. Self-attention layer takes key, query, and value as the input. On the other hand, convolutional neural networks (CNNs) are well-known for their ability to extract essential features effectively from local spatial hierarchies. This study proposes a combination of a multi-head attention (MHA) layer and a one-dimensional convolutional layer instead of FF for time-series data diagnostic. As the convolutional layer extracts essential features, the multi-head attention layer enhances the model's performance by directing each head to focus on distinct aspects of the data, thereby providing complementary information crucial for accurate fault classification.

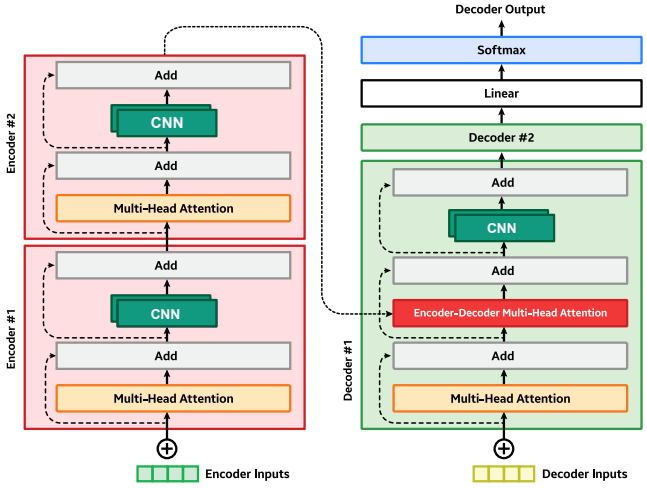

Fig. 4. A schematic of the transformer neural network structure including convolutional layers

Figure 4 represents a schematic of the proposed transformer neural network. A SoftMax activation layer is implemented at the output layer to yield a probability that each fault occurs within the system.

2.3 Training methodology

CNNs are more likely to experience overfitting than traditional ML models or simpler neural network architectures. One way to tackle this issue is to include dropout layers in the CNN structure. This will disable some of the connections between layers in the training process to achieve a generalized model. Another way is to define loss value criteria to terminate the training if the model has not improved in multiple consequence epochs. This research applies a dropout rate of 10-20% to the neural network structure. The loss was calculated using the categorical cross-entropy function, as shown in (4).

$$CCE(y,\hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j}\log(\hat{y}_{i,j}) \qquad (4)$$

Where N is the number of samples, C is the number of classes, y is a Boolean to show if the sample i belongs to class j, and $\hat{y}$ is the predicted probability that the sample i belongs to class j.

Classification accuracy, a fundamental performance metric in supervised learning, is calculated as the ratio of correctly predicted instances to the total number of instances evaluated. Formally, accuracy (A) is defined as:

$$A = \frac{1}{N}\sum_{i=1}^{N} l(y_i = \hat{y}_i) \qquad (5)$$

Where in (5), N denotes the total number of predicted class labels, and $l(\cdot)$ is an indicator function that returns 1 if the argument is true and 0 otherwise. This metric effectively quantifies the proportion of correct predictions made by the model, providing an intuitive measure of its performance.

A batch size of 256 was selected to gain consistent training, testing, and validation results. Considering the batch size, a learning rate of η =5e-5 is assumed for the training. The input data of the neural network is a window of input variables with the size w = 10 and created using past and present records of measured values.

Starting from epoch 70, an early stopping condition on the validation loss was applied to terminate the training process after 5-20 consecutive epochs without any improvement. The models were optimized by Adam optimizer. The Adam optimizer offers adaptive learning rates and efficient computation, which makes it suitable for large datasets and models with sparse gradients. Its bias correction, robustness to noisy gradients, and user-friendly hyperparameters ensure fast convergence and broad applicability in deep learning.

A summary of the model structures studied in this research is shown in Table 1.

Table 1. Summary of the model structures of Vanilla CNN, CNN, FF Transformer, and Convolutional Transformer

1. Vanilla CNN

#	layer type	unit/ filter	activation function
1	Conv1D	100	ReLU
2	Avg Pooling	-	-
3	Dropout*	10%	-
4	Conv1D	100	ReLU
5	Avg Pooling	-	-
6	Dense**	2	SoftMax

2. CNN

#	layer type	unit/ filter	activation function
1	Conv1D	100	ReLU
2	Conv1D	100	ReLU
3	Dropout	10%	-
4	Dense**	2	SoftMax

3. FF Transformer

#	layer type	unit/ filter	activation function
1	MHA	4	-
2	Dense**	100	ReLU
3	Dropout	10%	-
4	MHA	4	-
5	Dense**	100	ReLU
6	Dense	100	ReLU
7	Dense	2	SoftMax

4. Convolutional Transformer

#	layer type	unit/ filter	activation function
1	MHA	4	-
2	Conv1D	100	ReLU
3	Dropout	20%	-
4	MHN	4	-
5	Conv1D	100	ReLU
6	Dense**	2	SoftMax

* Dropout rate

** A flattened layer is applied before the dense layer

Table 2 provides an overview of the data split into train, validation, and test sets. The test data set uses units 10 and 16 to ensure that the model evaluates with a balanced data set and that there is no data leakage between train/validation and test splits. Of the remaining units, 80% and 20% were utilized for training and validation, respectively.

Table 2. An overview of the training, validation, and test datasets

Dataset	Unit	Fault Mode
training and validation	2 5	HPT
	11 14 15 18 20	HPT + LPT
test	10	HPT
	16	HPT + LPT

Data were normalized from 0 to 1 using a Min-Max scaler to avoid saturation in the activation function and achieve faster convergence. We used TensorFlow 2.13 for the implementation of our neural network models, running on a system with an AMD Ryzen 9 5950X CPU (16 cores, 32 threads), an NVIDIA GeForce RTX 3090 GPU (24 GB of GDDR6X memory), and 128 GB of DDR4 RAM at 3200 MHz.

3. RESULTS AND DISCUSSION

The K-Best feature selection was implemented to analyze the effect of each variable on fault classification. The K-Best score for each variable and the cumulative score are presented in Fig. 5. The results indicate that P50 contributes the most to capturing the variance of the output, and the LPT outlet temperature (T50) has the most negligible effect on capturing the output. In other words, the impact of T50 on fault classification is insignificant, and 13 variables are enough to predict the output. Notably, 14 out of 45 variables were selected as the model's input. In this case, the effect of degradation on T50 was not considerable and resulted in a lower K-Best score. However, based on the physics of the problem, we decided to keep T50 as the model's input.

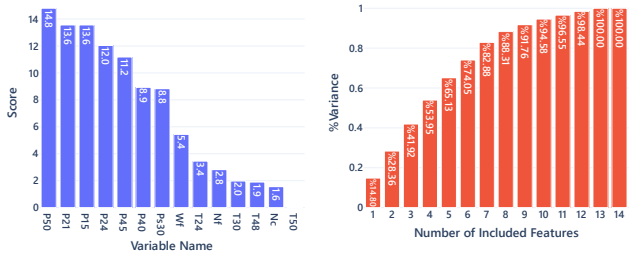

Fig. 5. K-Best feature selection results: variable scores (left) and cumulative score (right)

Figure 6 shows the result of the K-Means clustering. The optimal number of clusters was identified by the knee method. As shown, the input variables can be classified into 4 clusters. The variables in each cluster are expected to have a similar dynamic behavior. For example, the HPT outlet temperature (T48) and the T50 have similar dynamic behavior and can be assumed to be a cluster. Table 3 shows the selected features for the classification problem.

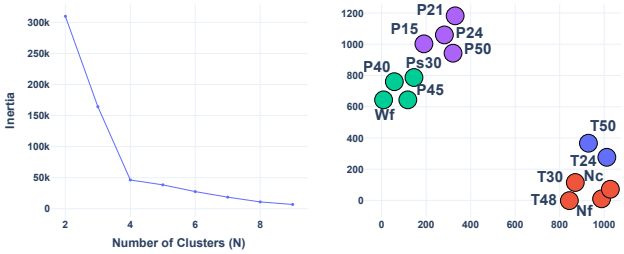

Fig. 6. K-Means inertia value based on the number of clusters (left) and K-Means for $n_c = 4$ (right)

Figure 7 presents training and validation loss curves. Reducing the learning rate resulted in smoother training curves and stable results. The model reaches the best accuracy around epoch 120. Starting from epoch 90, the training loss starts to fluctuate around the local minimum. No significant improvements were observed after 120 epochs, and the training was terminated to avoid overfitting. In the presence of a dropout layer, a lower validation loss is expected compared to the training loss.

Table 3. Description and notation of variables within the dataset based on Arias Chao et al. (2021)

#	Symbol	Description	Units
1	Wf	Fuel flow	kg/s
2	Nf	Physical fan speed	rpm
3	Nc	Physical core speed	rpm
4	T24	Total temperature at LPC outlet	K
5	T30	Total temperature at HPC outlet	K
6	T48	Total temperature at HPT outlet	K
7	T50	Total temperature at LPT outlet	K
8	P15	Total pressure in bypass-duct	kPa
9	P21	Total pressure at fan outlet	kPa
10	P24	Total pressure at LPC outlet	kPa
11	Ps30	Statics pressure at HPC outlet	kPa
12	P40	Total pressure at burner outlet	kPa
13	P50	Total pressure at LPT outlet	kPa
14	P45	Total pressure at HPT outlet	kPa

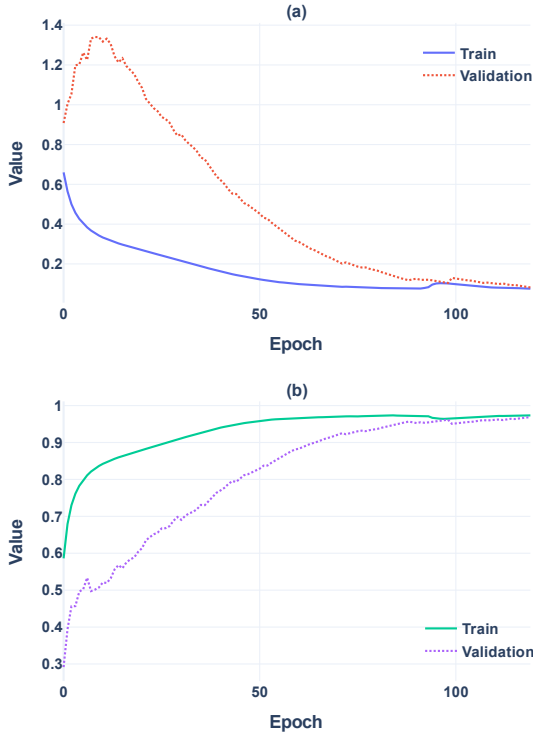

Fig. 7. Convolutional Transformer loss (a) and accuracy (b) curves for the training and validation datasets

Imbalanced classes in the training dataset cause a temporary increase in the validation loss at the preliminary stages of the training. The prediction results at the beginning are biased toward the class with a higher population. On the other hand, the categorical cross-entropy loss is sensitive to the probabilities assigned to the correct classes. If the model's confidence in its predictions fluctuates (even if predictions are mostly correct), the loss can increase while accuracy improves. This behavior was observed across all the models studied during this research. Training and validation loss/accuracy curves for Vanilla CNN, CNN, and FF Transformer are presented in Appendix A.

The validation and test accuracies prove these claims since both show remarkable results in fault classification. Figure 8 shows the confusion matrix for the transformer model with convolutional layers. As shown, imbalanced data for fault classes resulted in a considerable gap between Fault 1 and 2 accuracies for models 1-3. These models can identify Fault 1 with higher accuracy despite a higher population of classes with Fault 2. As mentioned, it has simple features and only corresponds to HPT fault, while Fault 2 is a combination of HPT and LPT failures.

Hence, a convolutional transformer (model 4) shows significant improvement in finding the complex correlation in the data considering the imbalance training/validation dataset. It is good to note that Fault 2 is a combination of Fault 1 and other faults and distinguishing one from another is challenging.

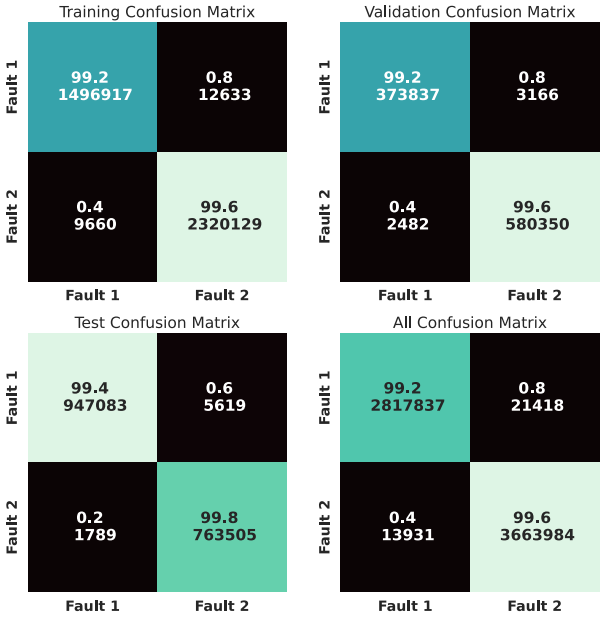

Fig. 8. Convolutional transformer confusion matrix for training, validation, testing, and total dataset

Interestingly, the accuracy gained from the test dataset is promising and higher than both training and validation in some cases. Considering that the test dataset is from separate turbofan engine units and was never used during the training season, it is another clue for the quality of training and evaluation. Implementing the dropout layer avoids overfitting and makes the model learn the pattern instead of overfitting the current dataset. In this case, the final model performs better in predicting the test dataset.

Figure 9 summarizes the overall accuracy of the evaluated modes in this study. CNN model lacks MaxPooling layers and has higher parameters compared to Vanilla CNN. In this case, the CNN model performs slightly better in accuracy. The transformer model with feed-forward layers improved overall accuracy by 0.8%.

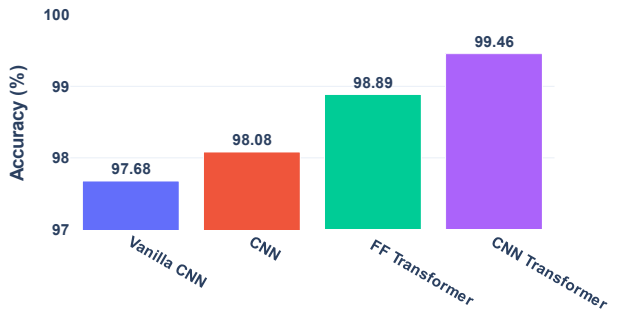

Fig. 9. Overall fault classification accuracy of Vanilla CNN, CNN, FF Transformer, and Convolutional Transformer neural networks

What is interesting here is the accuracy of 98.89% achieved by substituting convolutional layers by feed-forward and adding the attention mechanism. A combination of multi-head attention and convolutional layers improves spatial pattern recognition and results in around 99.5% overall accuracy in fault classification. Despite the highest accuracy in the Convolutional transformer model, the FF transformer offers a simple model structure with fewer parameters and respectful accuracy. However, compared to the Convolutional transformer, this structure fails to handle an imbalanced dataset.

4. CONCLUSIONS

Gas turbines play a vital role in transportation and energy systems. Modern engineering considers optimal design, efficient operation, sustainability, and safety simultaneously when manufacturing turbofan engines, with the progress of ML and artificial intelligence process optimization and maintenance going toward automation to balance safety and diagnostics costs. This research investigated the application of recent developments in NLP and adopted a minimal structure to provide a fault classifier for the gas turbine. The final platform fulfills the needs of AI assistants to identify the faults within the turbofan engines based on a snapshot of measured data in real-time. The transformer takes advantage of the multi-head attention mechanism and convolutional layers to investigate a horizon of information and capture the essential information. The results for the proposed model show an accuracy of 99.6% on the test dataset. In future work, a complete dataset of faults will be investigated to analyze the effect of each measurement on anomaly detection. While proposing a general model for the gas turbine diagnostic is challenging, providing more information will help the model recognize more patterns and distinguish the system behavior under different conditions. The outcome can be fine-tuned for distinct types of gas turbines for various kinds of applications.

REFERENCES

Arias Chao, M., Kulkarni, C., Goebel, K., and Fink, O. (2021). Aircraft Engine Run-to-Failure Dataset under Real Flight Conditions for Prognostics and Diagnostics. *Data*, *6*(1), Article 1. doi:10.3390/data6010005

Arpit, D., Kanuparthi, B., Kerg, G., Ke, N. R., Mitliagkas, I., and Bengio, Y. (2019). *h-detach: Modifying the LSTM Gradient Towards Better Optimization* (arXiv:1810.03023). arXiv. doi:10.48550/arXiv.1810.03023

Barrera, J. M., Reina, A., Mate, A., and Trujillo, J. C. (2022). Fault detection and diagnosis for industrial processes based on clustering and autoencoders: A case of gas turbines. *International Journal of Machine Learning and Cybernetics*, *13*(10), 3113–3129. doi:10.1007/s13042-022-01583-x

Cao, Q., Chen, S., Zheng, Y., Ding, Y., Tang, Y., Huang, Q., Wang, K., and Xiang, W. (2021). Classification and prediction of gas turbine gas path degradation based on deep neural networks. *International Journal of Energy Research*, *45*(7), 10513–10526. doi:10.1002/er.6539

Fentaye, A. D., Baheta, A. T., Gilani, S. I., and Kyprianidis, K. G. (2019). A Review on Gas Turbine Gas-Path Diagnostics: State-of-the-Art Methods, Challenges and Opportunities. *Aerospace*, *6*(7), Article 7. doi:10.3390/aerospace6070083

Fentaye, A. D., Zaccaria, V., and Kyprianidis, K. (2021). Aircraft Engine Performance Monitoring and Diagnostics Based on Deep Convolutional Neural Networks. *Machines*, *9*(12), Article 12. doi:10.3390/machines9120337

Losi, E., Venturini, M., Manservigi, L., Ceschini, G. F.,Bechini, G., Cota, G., and Riguzzi, F. (2022). Prediction of Gas Turbine Trip: A Novel Methodology Based on Random Forest Models. *Journal of Engineering for Gas Turbines and Power*, *144*(031025). doi:10.1115/1.4053194

Pascanu, R., Mikolov, T., and Bengio, Y. (2012). On the difficulty of training Recurrent Neural Networks. *30th International Conference on Machine Learning, ICML 2013*.

Sateesh Babu, G., Zhao, P., and Li, X.-L. (2016) *Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life* (S. B. Navathe, W. Wu, S. Shekhar, X. Du, X. S. Wang, and H. Xiong, Eds.; Vol. 9642, pp. 214–228). Springer International Publishing. doi:10.1007/978-3-319-32025-0_14

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). *Attention Is All You Need* (arXiv:1706.03762). arXiv. doi:10.48550/arXiv.1706.03762

Volponi, A. J. (2020). *Gas Turbine Parameter Corrections*. Springer International Publishing. doi:10.1007/978-3-030-41076-6

Xie, J., Sage, M., and Zhao, Y. F. (2023). Feature selection andfeature learning in machine learning applications for gas turbines: A review. *Engineering Applications of*

Artificial Intelligence, *117*, 105591. doi:10.1016/j.engappai.2022.105591

Yang, X., Cheng, K., Zhao, Q., and Wang, Y. (2023). Unknown fault detection for EGT multi-temperaturesignals based on self-supervised feature learning and unary classification. *Frontiers in Energy*, *17*(4), 527–544. doi:10.1007/s11708-023-0880-x

Yang, Z., Wong, P. K., Vong, C. M., Zhong, J., and Liang, J. (2013). Simultaneous-Fault Diagnosis of Gas Turbine Generator Systems Using a Pairwise-Coupled Probabilistic Classifier. *Mathematical Problems in Engineering*, *2013*(1), 827128. doi:10.1155/2013/827128

Zhou, D., Zhang, H., and Weng, S. (2015). A New Gas Path Fault Diagnostic Method of Gas Turbine Based on Support Vector Machine. *Journal of Engineering for Gas Turbines and Power*, *137*(102605). doi:10.1115/1.4030277

Appendix A: Confusion matrix and training/validation loss curves for Vanilla CNN, CNN, and FF transformer

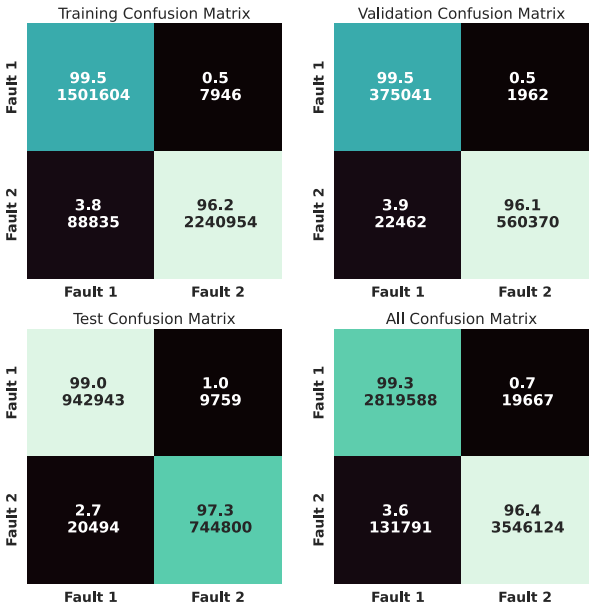

Fig. A1. Vanilla CNN confusion matrix for training, validation, testing, and total dataset

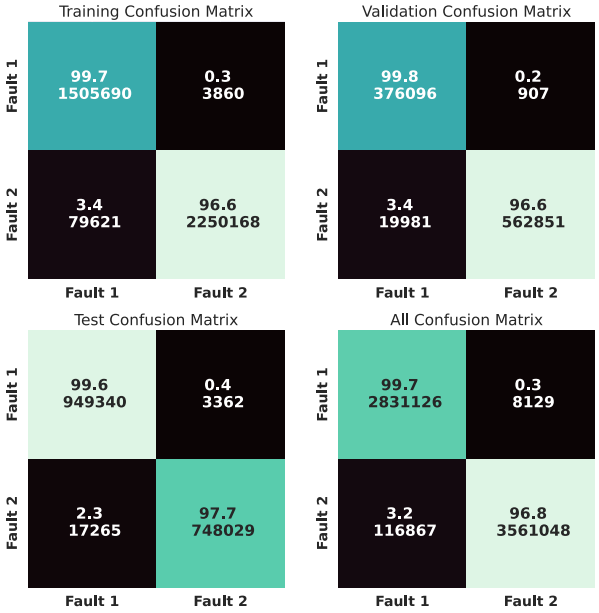

Fig. A2. CNN confusion matrix for training, validation, testing, and total dataset

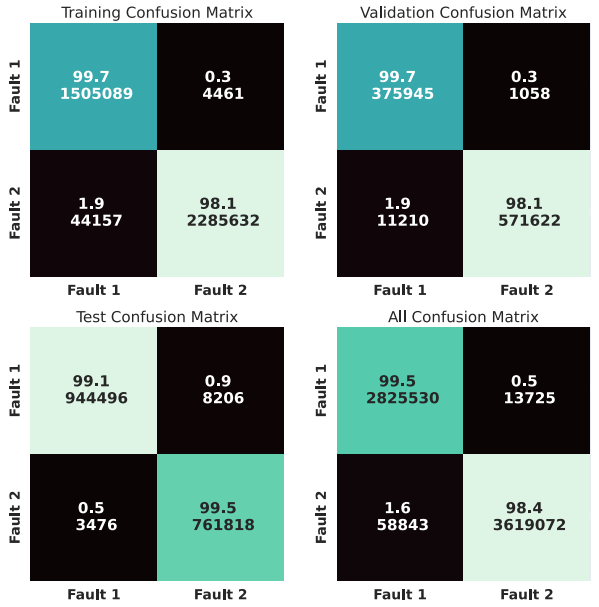

Fig. A3. FF transformer confusion matrix for training, validation, testing, and total dataset

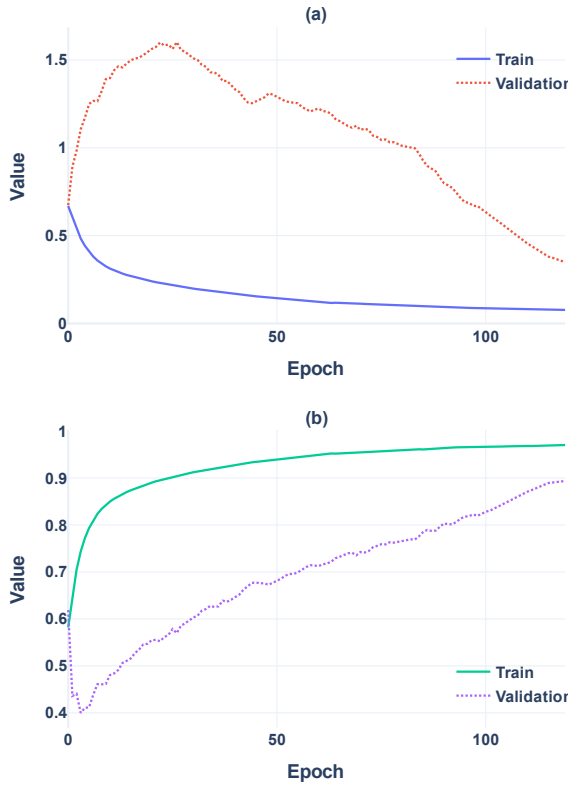

Fig. A4. Vanilla CNN loss (a) and accuracy (b) curves for the training and validation datasets

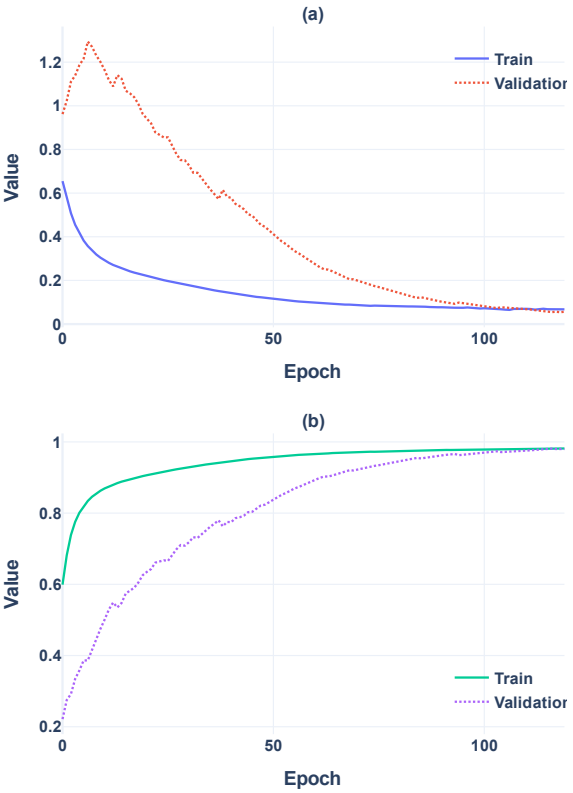

Fig. A5. CNN loss (a) and accuracy (b) curves for the training and validation datasets

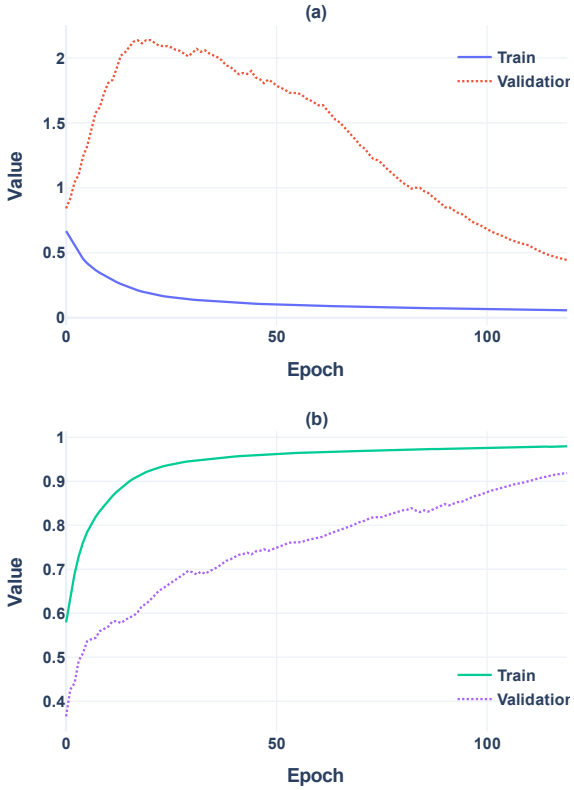

Fig. A6. FF Transformer loss (a) and accuracy (b) curves for the training and validation datasets