Engineering and Technology Journal e-ISSN: 2456-3358

Volume 11 Issue 03 March-2026, Page No.-9286-9294

DOI: 10.47191/etj/v11i03.23, I.F. – 8.595

© 2026, ETJ

Autoencoder-Based Anomaly Detection for Turbofan Engine Sensors Data

Amin Jainal Arivin[1], Suwanto Sanjaya[2*], Jasril[3], Yelfi Vitriani[4], Iis Afrianty[5]

[12345]Informatics Engineering, Faculty of Science and Technology, Sultan Syarif Kasim Riau State Islamic University

ABSTRACT Turbofan engines are an important part of aircraft that generate a large amount of multivariate sensor data during operation. The C-MAPSS FD001 dataset released by NASA accurately reflects turbofan engine degradation data. FD001 contains rows of data representing one operational cycle of a specific engine with information about the Engine ID, Cycle, operational settings, and 21 sensor readings that reflect the physical condition and performance of the engine, but these are not labeled, making it difficult to identify anomalies directly. Therefore, an automated method is needed that can learn normal patterns and identify data deviations accurately. This study developed an autoencoder model based on Mean Squared Error (MSE) reconstruction to find anomalies in the C-MAPSS FD001 dataset. This model was created with an architecture consisting of an input layer (15 neurons), a latent layer (8 neurons), and an output layer (15 neurons), and was trained using 20,631 normalized training data (train_FD001). Anomalies were determined based on a 95th percentile threshold of the MSE value distribution. Testing was performed on test data (test_FD001) using a similar process. Testing results on 13,096 test data points showed an average MSE value of 0.001823 with a standard deviation of 0.001032. From the 95th percentile threshold value of 0,003785, 131 data points, or about ±1%, were identified as anomalies. The low MSE value for most of the data indicates that the model can reconstruct normal data patterns well, while data with high MSE values can be identified as anomalies. This study confirms that autoencoders with error reconstruction are effective for detecting anomalies in unlabeled turbofan engine sensor data.

KEYWORDS: Autoencoder, Anomaly Detection, C-MAPSS FD001, Reconstruction Error, Turbofan Engine,

I. INTRODUCTION

Turbofan engines are an important part of aircraft because they have an additional fan at the front of the aircraft that can accelerate airflow in the ducts that pass through the core gas turbine engine [1]. Turbofan engines have many operational variables and sensors such as temperature, pressure, rotor speed, fuel flow, and other parameters[2]. The complexity of the sensor data in turbofan engines necessitates comprehensive and standardized simulation dataset to support engine condition monitoring research [3]. The C-MAPSS dataset contains simulations of turbofan engine operation, in which sensors record pressure, temperature, and other parameters throughout the engine's life cycle [4]. C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) is a commercial turbofan engine simulator developed by NASA to generate realistic run-to-failure data from turbofan engines under various operating conditions and degradation modes [5].

The C-MAPSS (Commercial Modular Aero Propulsion System) dataset is divided into four sub-datasets, namely FD001, FD002, FD003, and FD004, each of which has different characteristics [6]. FD001 is part of the Commercial Modular Aero Propulsion System Simulation (C-MAPSS) dataset, which presents multivariant run-to-failure data from turbofan engine sensors operating in a single condition with a single failure mode [7]. FD001 contains rows of data representing one operational cycle of a specific engine with information about the engine identity (Engine ID), operational cycle number, three operational settings (engine condition parameters), and 21 sensor readings that reflect the physical condition and performance of the engine [8]. The C-MAPSS FD001 dataset is widely used for scientific research, such as in the study [9] that uses the C-MAPSS dataset to develop multilayer perceptron (MLP**)** and Long Short-Term Memory (LSTM) in turbofan engines with a high level of accuracy. However, in this study, the C-MAPSS dataset is used as an object to detect anomalies that occur in the data. Although the C-MAPSS dataset is designed to represent realistic turbofan engine degradation, sensors often contain data anomalies or irregular data patterns due to non-ideal signal interference [10]. Data anomalies are events or values that deviate significantly from the general pattern in the dataset [11]. Anomalies are essentially points that differ from normal points without any chance of becoming normal points [12].

Anomalies generally reflect unusual events such as abnormal activity and extreme conditions that do not match the expectations of the analysis model [13]. In the context of turbofan engines, research [14] states that anomaly detection methods are effective for identifying different samples and separating abnormal behavior patterns in a single performance signal. Anomaly detection is the process of searching for data with behavior that differs from the norm [15]. According to research [16], anomaly detection is used to detect suspicious activity or abnormal cyber attacks on computer networks. Deep learning approaches such as autoencoders are able to recognize anomalies that are undetectable by traditional statistical methods by extracting latent patterns from machine operational data automatically and effectively in [17].

This study uses an autoencoder model to detect anomalies in turbofan engine sensor data (C-MAPSS FD001 dataset). Autoencoder is a type of artificial neural network architecture used in unsupervised learning [18]. Autoencoders are used in this study because they are capable of learning normal patterns in an unsupervised manner (without labels), producing reconstruction errors as indicators of deviations from normal conditions, and are relevant to the C-MAPSS dataset which has limited failure labels [19]. Research from [20] also states that the autoencoder model has the ability to learn normal pattern representations from multivariant sensor data without failure labels and can produce increased reconstruction errors if pattern deviations occur, making it effective for use as an early indicator of anomalies in complex turbofan engine systems. The autoencoder-based model allows for dimension reduction in response to unexpected changes in engine conditions [21]. Autoencoders work with the aim of reconstructing their input data. The encoder compresses the input data into a low-dimensional representation called latent space or code, while the decoder attempts to reconstruct the original input from the encoded representation [22]. With the ability to find hidden structures in the data, autoencoders can perform compression well and can also learn non-linear features [23]. Autoencoders are designed to produce more concise data representations and more important information by compressing and then decompressing the input data. Dimension reduction not only reduces the dimensions of the original data but also makes hidden features easier to understand [24]. In the study [17], autoencoders were used to detect anomalies occurring in vibration signals. The results obtained show that autoencoders are an effective and efficient solution for detecting anomalies in industrial machine vibration signals. With an accuracy of 94%, the results show that autoencoders are able to distinguish between normal and anomalous conditions with high accuracy.

This research was conducted with the aim of developing and evaluating an autoencoder model based on error reconstruction (MSE) to detect anomalies in unlabeled turbofan engine sensor data. The data was obtained from NASA, namely C-MAPSS FD001 (train_FD001). With 20,631 rows of data, with 20,631 rows of data, 26 column attributes including Engine ID, operational cycle number, three operational settings (engine operating conditions), and 21 sensor readings reflecting the physical condition of the turbofan engine. This research is expected to produce a model that can detect anomalies in unlabeled sensor data.

II. RESEARCH METHODOLOGY

A. Research Stages

The research methodology presents the stages of research that will be carried out in solving the problem in accordance with the objectives so that the results obtained are appropriate. The stages to be carried out are as follows:

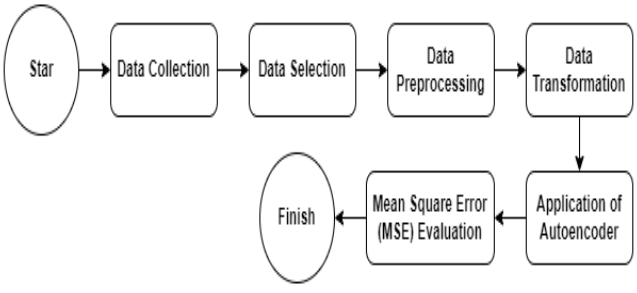

Figure 1 Research Stage

B. Data Collection

In this study, the data used was obtained from the NASA website, specifically the C-MAPSS turbofan engine degradation simulation https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/. C-MAPSS FD001 (train_FD001) consists of 20,631 data rows with 26 attribute columns covering Engine_ID, cycles, three operational settings (engine operating conditions), and 21 sensor readings reflecting the physical condition of the turbofan engine.

Table 1 Raw Data Excerpt from the C-MAPSS FD001 Dataset Showing Operational Parameters and Turbofan Engine Sensors

No	Engine ID	Cycle	Setting 1	…	Setting 3	Sensor 1	…	Sensor 21
0	1	1	-0.0007	…	100.0	518.67	…	23.4190
1	1	2	0.0019	…	100.0	518.67	…	23.4236
…	…	…	…	…	…	…	…	…
20629	100	199	-0.0011	…	100.0	518.67	…	23.0640
20630	100	200	-0.0032	…	100.0	518.67	…	23.0522

C. Data Selection

Selection of column attributes to be used in accordance with the requirements. The attributes used must be in line with the research objectives so that the selected data attributes are highly relevant. This selection is based on the research objective, which is to detect anomalies in turbofan engine sensors, so that the selection of attributes only uses sensor attributes because sensors are a reflection of the physical condition of the engine, while attributes such as Engine_ID, cycle, and Setting are not used in this study because they are not included in the engine sensors. This stage aims to ensure that the data is ready for analysis.

D. Data Preprocessing

Data Preprocessing is a crucial step to ensure data quality before it is used in the modeling process. This process focuses on data cleaning, which aims to eliminate missing values, duplicate data, and data attributes that do not vary in value (stagnant). With stagnant data attributes, the model can become complex or prone to overfitting. With stagnant data attributes, the model can become complicated or prone to overfitting [25]. Furthermore, data that does not vary will generally add noise and complexity without contributing [26].

E. Data Transformation

Data transformation aims to normalize feature values to achieve the same scale. In this study, min-max normalization plays an important role in data transformation because it can equalize the value scale between features so that model performance improves [27]. This process converts the original values into values in the range of 0 to 1, without changing the relative distribution pattern between data [28].

Differences in scale between attributes can affect machine learning modeling because high scale values will dominate and low scale values will be ignored [29]. The following is the min-max normalization equation:

$$X_{normal} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Explanation:
X = original value
$Xmin$ = minimum value of the column or feature
$Xmax$ = maximum value of the column or feature
$Xnormal$ = normalized value.

F. Application of Autoencoder

Autoencoders are a type of neural network architecture that can be used in both unsupervised and supervised learning paradigms [30]. Autoencoders work by performing data reduction (Dimension Reduction) and then reconstructing data that is considered important in a pattern or model, the model in the encoder section is tasked with extracting important features from motifs stored in latent space, which are then processed by the decoder to restore the data for use as pattern recognition input in the neural network [31]. Choosing a model architecture for dimension reduction can affect the model results. In research [32], it is stated that reducing 25%-50% of the dimensions (laten space) of the input can achieve good reconstruction results while maximizing accuracy. By reducing the data dimensions, autoencoders also improve model performance on specific tasks such as classification, anomaly detection, and data visualization in low-dimensional space [33].

Training a model to divide a dataset is one strategy in machine learning to maintain a balanced learning process and evaluate model performance, based on research [34] the 80:20 ratio (80% training and 20% validation) is used to ensure that the model obtains a sufficient amount of data to learn patterns and characteristics of the data while also providing data that has never been seen by the model as a basis for performance evaluation. In addition, the selection of optimize and batch_size greatly affects the autoencoder training process. The use of Adam optimization is widely applied in deep learning models due to its ability to accelerate the convergence process and improve model accuracy [35]. Meanwhile, the use of a batch_size of 64 has been proven to maintain a balance between training speed, gradient stability, and model accuracy, as shown in the study [36] where the use of a batch_size of 64 in the LSTM (Long Short-Term Memory) model was proven to produce a lower Mean Square Error value than the others, thus demonstrating that this batch size is capable of providing stable convergence and optimal model performance.

A study written by [17] states that in the encoding stage, input data X is projected into the model, then produces data h obtained after the encoding stage. The encoding process is described in the following equation:

$$h = f(x) = f(W_1 X + b_1) \quad (2)$$

Explanation:
$W1$ = is the weight matrix between the input layer and the hidden layer, and
$b1$ = is the bias vector.
f(x) = is the activation function used for non-linear transformation

Next is decoding, which aims to return the h data to its original form and reconstruct X^d:

$$X^d = g(x) = g(W_2 h + b_2) \quad (2.1)$$

Explanation:
$W2$ = is the weight matrix between the hidden layer and the output layer, and
g(x) = is the activation function
$b2$ = is the bias vector
The function g(x) = can be a nonlinear transformation or an affine transformation function

G. Mean Square Error (MSE) Evaluation

Mean Square Error (MSE) will detect anomalies by calculating the reconstruction error from the sensor data [37]. The anomaly detection mechanism begins by calculating the reconstruction error, which is defined as the difference between the original input and its reconstructed version [38]. All reconstruction errors in the dataset will be collected to determine the dynamic threshold [39]. The reconstruction error will be averaged squared, which becomes a measure of the model's prediction accuracy and becomes the MSE or reconstruction error [40]. At this stage, the MSE value is calculated for each sensor individually using the equation:

$$MSE_i = \frac{1}{d}\sum_{j=1}^{d}(x_{ij} - \hat{x}_{ij})^2 \quad (3)$$

Explanation:
x_{ij}= the original value of feature j in sample i,
$\hat{x}_{ij}$ = the reconstructed value of feature j, and
d = the number of features in each sample.

After obtaining the error reconstruction results, a percentile is selected for the threshold on the error reconstruction value as an anomaly indicator. This approach is taken because the anomaly label is not explicitly available [41]. The selection of a percentile is important in detecting anomalies. As in study [42], the use of percentiles such as 95th, 97.5th, and 99th has strong reasons. The 95th percentile is more sensitive to minor anomalies, the 97.5th percentile has a stricter limit than the 95th percentile, and the 99th percentile is even stricter than the previous ones and is not sensitive to minor anomalies but is more sensitive to extreme error reconstruction values. The use of a 99% threshold or 3-sigma rule is too conservative because it can only detect extreme anomalies, potentially ignoring minor anomalies that are actually important in the context of predictive maintenance. Therefore, the 95% percentile is used as a trade-off between detection sensitivity and false alarm rate, where this threshold is still able to increase sensitivity to early deviation patterns without generating excessive false alarms [43]. The following is the equation for determining the percentile threshold:

$$i = \frac{P}{100} \times n \quad (4)$$

Explanation:
i: is the data index position at the Pth percentile
P: is the percentile value used (percentile 95)
n: is the sample size of the data

The value occupying index position (i) is then set as the threshold. Once the threshold value is obtained, the detection rule is, if the MSE value is lower than the threshold value, it is considered normal; if the MSE value is higher than the threshold value, it is considered an anomaly.

III.RESULT AND DISCUSSION

A. Data Selection

Data selection was based on the research objectives, which were to use only attributes Sensor 1 to Sensor 21.

Table 2 Selection Results Data from the C-MAPSS FD001 Dataset

No	Sensor 1	Sensor 2	Sensor 3	…	Sensor 20	Sensor 21
0	518.67	641.82	1589.70	…	39.06	23.4190
1	518.67	642.15	1591.82	…	39.00	23.4236
…	…	…	…	…	…	…
20629	518.67	643.23	1605.26	…	38.29	23.0640
20630	518.67	643.85	1600.38	…	38.37	23.0522

Other attributes such as Engine_ID, cycle, and Settings 1–3 were not used in this study because they do not reflect the physical condition of the engine and do not provide information about the physical condition of the engine.

B. Data Preprocessing

At this stage, the data will be cleaned of missing values, duplicate data, and stagnant values. The results show that the data has no missing values or duplicate data, but the data attributes are detected to have stagnant values in the Sensor1, Sensor5, Sensor10, Sensor16, Sensor 18, and Sensor19 data attributes.

Table 3 Data Preprocessing Results with Stagnant Feature Elimination

No	Sensor 1	Sensor 5	Sensor 10	Sensor 16	Sensor 18	Sensor 19
0	518.67	14.62	1.3	0.03	2388	100.0
1	518.67	14.62	1.3	0.03	2388	100.0
…	…	…	…	…	…	…
20629	518.67	14.62	1.3	0.03	2388	100.0
20630	518.67	14.62	1.3	0.03	2388	100.0

Based on Table 3, stagnant data must be removed because it will interfere with the performance of the autoencoder model and is prone to overfitting. Therefore, the remaining data attributes used for this study are 15 attributes. Even though some attributes have been removed, the number of data rows remains the same, namely 20,631 rows, with int64 and float64 data types.

C. Data Transformation

At this stage, the data will be transformed using min-max scaling normalization in accordance with equation (1). The original data values will be converted to values in the range of 0 to 1.

Table 4 Data Transformation Results Using Min–Max Normalization

No	Sensor 2	Sensor 3	Sensor 4	…	Sensor 20	Sensor 21
0	0.183735	0.406802	0.309757	…	0.713178	0.724662
1	0.283133	0.453019	0.352633	…	0.666667	0.731014
…	…	…	…	…	…	…
20629	0.608434	0.746021	0.747468	…	0.116279	0.234466
20630	0.795181	0.639634	0.842167	…	0.178295	0.218172

Changes in value scales are important in running the model. If there are differences in scale between attributes, this can affect machine learning modeling, causing higher value scales to become dominant.

D. Application of Autoencoder

At this stage, the data used is the result of previous normalization, which consists of 15 attributes. The data will be divided into two parts, namely 80% training data and 20% validation data. In the autoencoder training process, the model is trained to learn patterns, and validation is used so that the data is not excessive during the training process (overfitting).

The autoencoder architecture used consists of 50% of the input amount, with the model using an input layer architecture (15 neurons), latent layer (8 neurons), and output layer (15 neurons). The training process uses Adam optimization, with a batch_size of 64 and a maximum number of epochs of 500 and is equipped with early stopping to automatically stop training when the model's performance no longer improves.

At the encoder stage, the input data is transformed into a more concise representation in the latent space using equation (2). Next, the decoder reconstructs the data in the latent space into an output that is the same as the initial input, which runs according to equation (2.1).

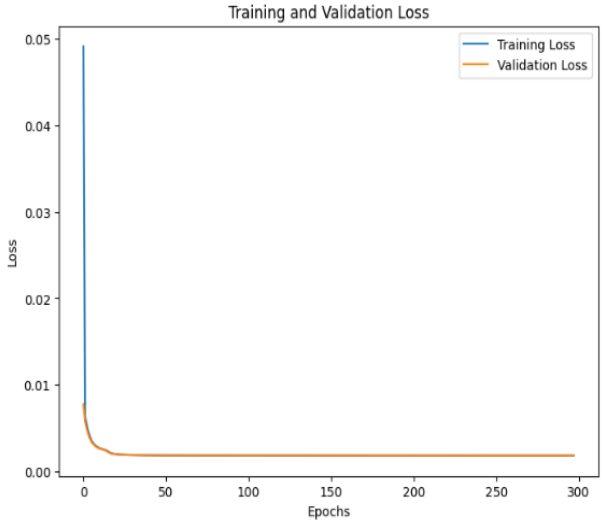

Figure 2 Architecture Training Results Graph

Figure 2, shows a consistent and stable loss reduction pattern, but the final loss value tends to be slightly lower than the

training loss, with the gap between the training and validation losses remaining small throughout the training process. This architecture can be considered more optimal because it produces low reconstruction loss with maintained stability.

E. Mean Square Error (MSE) Evaluation

After the model is processed, MSE will calculate the resulting reconstruction. Next, MSE will be run using equation (3) to determine the error result (reconstruction error) and determine normal/anomalous data using a 95th percentile threshold, with equation (4).

Table 5 Statistical Results of Training Data Error Reconstruction

Description	Value
Count	4127.
Mean	0.001848
Std	0.001057
Min	0.000124
25%	0.001085
50%	0.001648
75%	0.002413
max	0.013858

Table 5 shows that the statistical results of training data error reconstruction indicate good reconstruction capabilities. The relatively low mean value of 0.001848 indicates that most of the data can be accurately reconstructed by the model. In addition, the standard deviation value, which is not too large, reflects a relatively narrow error distribution, while the small maximum value indicates that only a few samples have high reconstruction errors. These findings prove that the input layer (15 neurons), latent layer (8 neurons), and output layer (15 neurons) autoencoder architecture is capable of maintaining important data characteristics in a stable and consistent manner, so that this configuration can be declared as the optimal architecture in this study.

Error evaluation of the reconstruction using the Mean Square Error (MSE) method was performed on each row of data by calculating the difference between the original sensor value and the reconstruction value produced by the model. These differences were squared and then averaged to obtain the error value for each data sample. The reconstruction error calculation process covered all sensor attributes used, namely Sensor2 to Sensor21. Thus, a single MSE value was obtained, indicating the level of reconstruction error in each row of data. Next, anomaly detection is performed by setting a threshold based on the 95th percentile of the MSE value distribution. The selection of the 95th percentile aims to identify small deviations in the reconstruction data, as this percentile can separate a small portion of high error values from most data that behaves normally. The calculation results show that the threshold value based on the 95th percentile is 0.003796. Reconstruction error values that exceed this threshold are categorized as anomalies, while values below the threshold are considered normal data.

Table 6 Results of Reconstruction and Anomaly Detection in Training Data

No	Sensor 2	…	Sensor 21	Error Reconstruction	Anomaly
0	0.000108	…	0.000231	0.001752	False
1	0.000010	…	0.000564	0.001685	False
2	0.000411	…	0.007063	0.002840	False
…	…	…	…	…	…
20628	0.000311		0.010361	0.005368	True
20629	0.000049		0.002046	0.003456	False
20630	0.000004		0.000353	0.001265	False

Based on Table 6, the 95th percentile threshold is set at 0.003796 as the dividing line between normal and anomalous data. Of the 20,631 training data, 207 rows (±1%) were detected as anomalies. For example, data row 20628 was detected as an anomaly with a reconstruction error value (0.005368) that exceeded the threshold and was categorized as an anomaly, while values below the threshold were considered normal. Setting this percentile threshold allows for sensitive identification of data that deviates from the normal pattern, in accordance with the principle of autoencoder-based anomaly detection.

This shows that autoencoders are capable of detecting sensor data that deviates from normal data characteristics based on the magnitude of the reconstruction error. Lines with high MSE values indicate changes or deviations in machine sensor patterns that cannot be reconstructed properly by the model, and are therefore marked as anomalies.

F. Model Testing

The model will be tested using the test_FD001 data. This data is not used in the model training process but is new data for testing the previously trained model. This is done to determine whether the previously trained model is capable of detecting anomalies when using new data. The number of rows in the test_FD001 data is 13,096 with 26 columns of attributes.

The initial stage involves data selection, data preprocessing, and data transformation. In the data selection stage, column attributes are selected to suit the research objectives so that the selected data attributes are highly relevant. Data preprocessing is carried out with the aim of removing or correcting missing values, duplicate data, and data attributes that do not have value variations (stagnant). At this stage, the data was found to have 6 stagnant attributes. In the data

transformation stage, min-max normalization is performed to convert the original values into values in the range of 0 to 1. The purpose of this process is to make the data recognizable by the model. The data, which initially consisted of 26 attributes, was reduced to 15 due to 5 unused attributes (Engine_ID, Cycle, and 3 operational settings) and 6 attributes with no variation in values, which were identified during data preprocessing and removed.
Next, the model will reconstruct (encoder) the data, where the input will be compressed into a latent space, then the data will be returned (decoder) to its original dimension using the previously trained model.

Table 7 Error Reconstruction Statistics (MSE) of Autoencoder Results from Test Data

Description	Value
Count	13096
Mean	0.001823
Std	0.001032
Min	0.000044
25%	0.001073
50%	0.001612
75%	0.002349
max	0.010975

Based on Table 7, the statistical results of error reconstruction in the test data show a mean value of 0.001823 with a standard deviation of 0.001032, indicating that most of the reconstruction errors are within a relatively narrow range. The maximum value of 0.010975 indicates that a small portion of the data has high errors, but the majority of the data is below the third quartile (0.002349). This condition shows that the model is able to maintain stable reconstruction performance on the test data. Compared to the error reconstruction statistics results on the training data (Table 5), the Mean and standard deviation values on the test data are slightly lower, and the maximum error value is also smaller. This difference indicates that the model has good generalization capabilities because the error distribution on the test data is more controlled than on the training data.

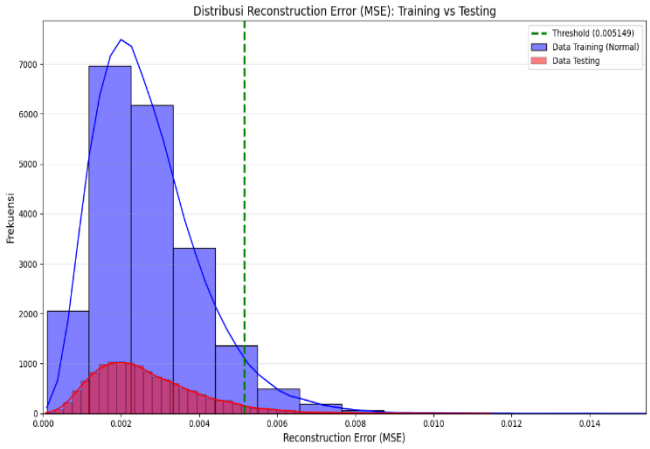

Figure 3 Training vs. Testing Histogram Chart

Based on Figure 3, the histogram shows that most of the MSE values in the training data (blue) are concentrated in the low range (around 0.001–0.003), indicating that the autoencoder is able to reconstruct normal patterns well. The distribution of the testing data (red) has a similar pattern but forms a longer right tail, indicating that there are a small number of samples that deviate from the normal pattern. The 95th percentile threshold line (green vertical) cuts off the tail of the distribution so that the testing data on the right side of the threshold is classified as an anomaly. This pattern confirms that the detected anomalies are statistical deviations from the nominal behavior of turbofan engines.
The MSE reconstruction error value is obtained by calculating the difference between the original sensor data and the model reconstruction results, thereby producing an error value for each sensor. All error values are then compared with the 95th percentile threshold as the anomaly detection criterion. Based on the calculations, the 95th percentile threshold value used in this study is 0.003785.

Table 8 Results of Error Reconstruction and Anomaly Detection in Test Data

No	Sensor 2	…	Sensor 21	Error Reconstruction	Anomaly
0	0.000165	…	0.000199	0.001891	False
1	0.000046	…	0.000406	0.004330	False
…	…	…	…	…	…
8	0.201807	…	0.000015	0.004012	True
…	…	…	…	…	…
13094	0.000041	…	0.004437	0.001369	False
13095	0.000012	…	0.000246	0.001572	False

Based on Table 8, the threshold value is set at 0.003785. Data rows with reconstruction error values exceeding this threshold are classified as anomalies, while data with values below the threshold are categorized as normal. Of the 13,096 test data, 131 data rows (±1%) were detected as anomalies because they exceeded the threshold. For example, row 8 with a reconstruction error value of 0.004012 was detected as an anomaly because the reconstruction error value exceeded the specified threshold, while the other rows remained in the normal category according to the anomaly detection criteria.
The model testing results show that the model is capable of identifying a number of anomalous data in the test_FD001 dataset, indicating that the model has good generalization capabilities in detecting deviation patterns in new machine sensor data.

IV. CONCLUSION

Based on the results of the research that has been conducted, it can be concluded that the autoencoder model with a input layer (15 neurons), a latent layer (8 neurons), and an output layer (15 neurons) architecture was successfully used and tested as a way to detect anomalies in turbofan engine sensor data using the unlabeled C-MAPSS FD001 dataset. Data preprocessing, which includes data cleaning, removal of stagnant attributes, and min-max normalization, has proven to be effective in producing acceptable data for modeling, enabling the model to learn the machine's operational patterns well. The results of model training show that the autoencoder is capable of reconstructing normal data, as evidenced by the reconstruction error statistics on the training data with an average MSE of 0.001848 and a standard deviation of 0.001057. These values indicate that most of the data can be reconstructed stably and consistently. The use of a threshold with a 95th percentile approach resulted in a threshold value of 0.003796, which serves as a separator between normal and anomalous data. Testing the model with the test_FD001 dataset consisting of 13,096 data points resulted in an average MSE of 0.001823 with a standard deviation of 0.001032, which is very similar to the results on the training data. This shows that the model has good generalization capabilities and does not suffer from overfitting. Based on the 95th percentile threshold value of 0.003785, the model successfully detected 131 anomalous data points (±1%) from the total test data, indicating that a small number of data points have sensor patterns that differ from the characteristics of normal data. This research falls under the category of unsupervised anomaly detection, as the FD001 dataset does not provide anomaly annotations for each observation cycle. Therefore, the existence of anomalies is not validated through reference labels, but rather determined using statistical criteria in the form of reconstruction error values. Samples that produce errors above the 95% percentile limit are treated as indications of deviations from the operational characteristics of the machine that are considered normal. Thus, these detection results represent potential deviations in system behavior, not actual failures**.** For further research, it is recommended to explore the autoencoder architecture in greater depth, test variations in percentile threshold values, and apply it to the FD002, FD003, and FD004 sub-datasets to improve model generalization. Additionally, integrating anomaly detection results with Remaining Useful Life (RUL) predictions is recommended so that the system is not only capable of detecting deviations but also predicting machine failure times more comprehensively.

REFERENCES

1. N. D. Retnowati, D. Nugraheny, and H. Sunaryo, "Pemodelan Dan Simulasi Cara Kerja Turbofan," *Pros. Semin. Nas. Sains Teknol. dan Inov. Indones.*, vol. 3, no. November, pp. 24–25, 2021, doi: 10.54706/senastindo.v3.2021.136.
2. Y. Duan, T. Zhang, and D. Shi, "Anomaly Detection and Remaining Useful Life Prediction for Turbofan Engines with a Key Point-Based Approach to Secure Health Management," *Sensor*, vol. 24, 2024.
3. Y. Wang, M. Ragab, Y. Hou, and Z. Chen, "Deep Domain Adaptation for Turbofan Engine Remaining Useful Life Prediction : Methodologies , Evaluation and Future Trends," *arXiv*, pp. 1–20, 2025.
4. S. M. Elsherif, B. Hafiz, M. A. Makhlouf, and O. Farouk, "OPEN A deep learning-based prognostic approach for predicting turbofan engine degradation and remaining useful life," *Sci. Rep.*, pp. 1–31, 2025.
5. F. Wang, A. Liu, C. Qu, R. Xiong, and L. Chen, "A Deep-Learning Method for Remaining Useful Life Prediction of Power Machinery via Dual-Attention Mechanism," *Sensor*, vol. 25, pp. 1–19, 2025.
6. A. Kara, "A Hybrid Prognostic Approach Based on Deep Learning for the Degradation Prediction of Machinery," *Sak. Univ. J. Comput. Inf. Sci.*, vol. 4, no. 2, pp. 216–226, 2021, doi: 10.35377/saucis.04.02.912154.
7. Y. Liu, G. Zhang, and J. Li, "Temporal Convolutional Networks Applied to Remaining Useful Life Prediction of Turbofan Engine," *J. Comput.*, vol. 32, no. 3, pp. 69–85, 2021, doi: 10.3966/199115992021063203006.
8. C. Peng, Y. Chen, W. Gui, Z. Tang, and C. Li, "Remaining useful life prognosis of turbofan engines based on deep feature extraction and fusion," *Sci. Rep.*, no. 0123456789, pp. 1–14, 2022, doi: 10.1038/s41598-022-10191-2.
9. A. Chaoub and C. Cerisara, "Learning representations with end-to-end models for improved remaining useful life prognostics," *arXiv*, vol. 10, 2021.
10. H. D. Nguyen, K. P. Tran, S. Thomassey, and M. Hamad, "Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management," *Int. J. Inf. Manage.*, vol. 57, 2021.
11. G. A. Ohyver, J. Tji, D. Trisnawarman, S. Tiatri, and D. A. Ohyver, "Analisis data keuangan menggunakan isolation forest untuk meningkatkan akurasi deteksi anomali financial data analysis using isolation forest to improve anomaly detection accuracy," *J. Inf. Technol. Comput. Sci. Vol.*, vol. 8, no. 2022, pp. 1854–1860, 2025.
12. Milka Wijayanti Sunarto, Dendy Kurniawan, Edy Siswanto, and Haris Ihsanil Huda, "Deteksi Anomali Menggunakan Extended Isolation Forest (Eif)," *Tek. J. Ilmu Tek. dan Inform.*, vol. 1, no. 2, pp. 96–111, 2023, doi: 10.51903/teknik.v1i2.324.
13. I. M. S. T. Wibawa and A. A. istri N. E. Karyawati, "Isolation Forest dengan Exploratory Data Analysis pada Anomaly Detection untuk Data Transaksi," *J. Nas. Teknol. Inf. dan Apl.*, vol. 1, pp. 803–810, 2023.
14. Y. Ma, X. Zhu, J. Lu, P. Yang, and J. Sun, "Construction of Data-Driven Performance Digital Twin for a Real-World Gas Turbine Anomaly Detection Considering Uncertainty," *MDPI*, 2023.
15. D. R. K. Saputra, Y. V. Via, and A. N. Sihananto, "Deteksi Anomali Menggunakan Ensemble Learning Dan Random Oversampling Pada Penipuan Transaksi Keuangan," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 3, pp. 2779–2788, 2024, doi: 10.23960/jitet.v12i3.4910.
16. M. Fadhlurrohman, A. Muliawati, and B. Hananto, "Analisis Kinerja Intrusion Detection System pada

Deteksi Anomali dengan Metode Decision Tree Terhadap Serangan Siber," *Ilmu Komput. Agri-Informatika*, vol. 8, no. 2, pp. 90–94, 2021.
17. Z. Umari and J. Supardi, "Deteksi Anomali Sinyal Vibrasi pada Mesin Industri Menggunakan Autoencoder di PT . Pusri Palembang Anomaly Detection of Industrial Machine Vibration Signals Using Autoencoder at PT . Pusri Palembang," vol. 4, no. 12, pp. 737–746, 2024.
18. D. Prasetyawan and R. Gatra, "Analisis Cluster untuk Pengelompokan Kemampuan Penguasaan ICT Menggunakan K-Means dan Autoencoder," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 10, no. 2, pp. 145–157, 2025, doi: 10.14421/jiska.2025.10.2.145-157.
19. A. Garg, W. Zhang, J. Samaran, R. Savitha, S. Member, and C. Foo, "An Evaluation of Anomaly Detection and Diagnosis in Multivariate Time Series," *arXiv*, 2021, doi: 10.1109/TNNLS.2021.3105827.
20. Q. Xiao, S. Shao, and J. Wang, "Memory-augmented Adversarial Autoencoders for Multivariate Time-series Anomaly Detection with Deep Reconstruction and Prediction," *arXiv*, 2021.
21. W. Priatna, S. Yulianto, J. Prasetyo, S. Wijono, E. Maria, and D. Manongga, "Deteksi Anomali dalam Penipuan E-commerce Menggunakan Hybrid Autoencoder-Transformer Frameworks," *JEPIN (Jurnal Edukasi dan Penelit. Inform.*, vol. 11, no. 1, pp. 33–40, 2025.
22. P. W. Kurniawan S, Y. Kristian, and J. Santoso, "Pemanfaatan Deep Convulutional Auto-encoder untuk Mitigasi Serangan Adversarial Attack pada Citra Digital," *J-Intech*, vol. 11, no. 1, pp. 50–59, 2023, doi: 10.32664/j-intech.v11i1.845.
23. F. S. Nugraha and H. F. Pardede, "Autoencoder untuk Sistem Prediksi Berat Lahir Bayi," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 2, pp. 235–244, 2022, doi: 10.25126/jtiik.2022923868.
24. D. Prasetyawan, A. Mulyanto, and R. Gatra, "Pemetaan Lintasan Karir Alumni Berdasarkan Analisis Cluster: Kombinasi K-Means dan Reduksi Dimensi Autoencoder," *Edumatic J. Pendidik. Inform.*, vol. 9, no. 1, pp. 198–207, 2025, doi: 10.29408/edumatic.v9i1.29713.
25. L. D. Yunita, E. Utami, and A. Yaqin, "Pengolahan Data Sensor Gerak Ponsel untuk Klasifikasi Karakteristik Mengemudi," vol. 12, pp. 181–189, 2023.
26. Iqbal, N. Hidayat, D. P. Gevano, and A. P. R. Ilahi, "Segmentasi Pelanggan Menggunakan K-Means Clustering Berdasarkan Data Kepribadian dan Pola Konsumsi," vol. 6, no. 5, pp. 3914–3924, 2025.
27. A. Ambarwari, Q. J. Adrian, and Y. Herdiyeni, "Analisis Pengaruh Data Scaling Terhadap Performa Algoritme Machine Learninguntuk Identifikasi Tanaman," vol. 1, no. 10, pp. 117–122, 2021.
28. M. Fadilah and L. Tanti, "Klasifikasi Jenis Usaha Mikro Kecil Menengah (UMKM) Berdasarkan Skala Usaha MenggunakanAlgoritma C4.5," vol. 2, no. 1, pp. 814–827, 2025.
29. P. P. Allorerung, A. Erna, M. Bagussahrir, and S. Alam, "Analisis Performa Normalisasi Data untuk Klasifikasi K-Nearest Neighbor pada Dataset Penyakit," vol. 9, no. 3, pp. 178–191, 2024.
30. F. Rizal, W. Sri, and N. Muhammad, "Integrasi Convolutional Autoencoder dengan Support Vector Machine untuk Klasifikasi Varietas Integration of Convolutional Autoencoder with Support Vector Machine for Almond Varieties Classification," *J. Tek. Inform. dan Sist. Inf.*, vol. 11, no. April, pp. 63–77, 2025.
31. M. F. Dzulqarnain, A. Fadlil, and I. Riadi, "Implementasi CNN berbasis Autoencoder pada Klasifikasi Pola Batik," *Proceeding Informatics Collab. Dessimenation Meet. Infocoding 2025*, pp. 189–194, 2025.
32. A. V. Chikkankod and L. Longo, "On the Dimensionality and Utility of Convolutional Autoencoder ' s Latent Space Trained with Topology-Preserving Spectral EEG Head-Maps," pp. 1042–1064, 2022.
33. C. Hudaya, A. Gunawan, B. W. Tri, and P. A. Darat, "Penerapan Arsitektur JST Dalam Deep Learning Untuk Meningkatkan Akurasi Klasifikasi Gambar Dengan Autoencoder," *Semin. Nas. ForteiRegional 7*, vol. 7, no. 1, pp. 203–210, 2025.
34. G. Sarker, "Transfer Learning-Based Convolution Neural Network for Differentiation between Benign and Malignant Cancer Cells Using MobileNetV2," vol. 12, no. 7, pp. 24–30, 2025.
35. W. R. N. Rifa, D. Erwanto, and I. Yanuartanti, "Evaluasi Kinerja CNN dengan Optimizer RMSprop, Adam dan SGD dalam Klasifikasi Penyakit Daun Anggur," vol. 10, no. 1, pp. 91–104, 2025.
36. E. Nurmawati, R. Abaysa, and R. A. Putra, "Optimalisasi Portofolio Saham Syariah Berbasis Prediksi Menggunakan Long Short-Term Memory (LSTM)," vol. 10, no. 2, pp. 464–475, 2025, doi: 10.30591/jpit.v9ix.xxx.
37. S. Ryu, B. Jeon, H. Seo, M. Lee, J. Shin, and Y. Yu, "Development of Deep Autoencoder-Based Anomaly Detection System for HANARO," vol. 55, no. 2, pp. 475–483, 2023, doi: 10.1016/j.net.2022.10.009.
38. V. Claudia, J. Kaunang, N. Alamsyah, T. P. Yoga, and A. Hendra, "Optimized Deep Autoencoder With L1 Regularization and Dropout for Anomaly Detection in 6G Network Slicing," vol. 22, no. 2, pp. 189–194, 2025.
39. N. Arif, U. Mercu, and B. Yogyakarta, "Deteksi Anomali Harga Bitcoin Menggunakan Variational Autoencoder dan Extreme Value," *JITET (Jurnal Inform. dan Tek. Elektro Ter.*, vol. 13, no. 2, pp. 709–716, 2025.
40. K. Aviantoro and Y. Darnita, "Implementasi Wiener, Contrast Stretching, Sharpening Filter Pada Citra Semangka Menggunakan Mse,Rmse, Dan Psnr," *Djtechno J. Teknol. Inf.*, vol. 5, no. 2, pp. 195–205, 2024, doi: 10.46576/djtechno.v5i2.4613.
41. N. Alamsyah, H. F. Rahmani, and W. Erpurini, "Fine-Tuned Autoencoder Neural Network for Anomaly Detection in Accounting Transactions," vol. 27, no. 2, pp. 65–73, 2025.
42. A. Shpigler, N. Kolet, S. Golan, E. Weisbart, and A. Zaritsky, "Anomaly detection for high-content image-based phenotypic cell profiling," vol. 16, no. 11, p. 101429, 2025, doi:

10.1016/j.cels.2025.101429.
43. Z. Sun, W. Ye, and Y. Mao, "A Hybrid CNN – LSTM – Attention Mechanism Model for Anomaly Detection in Lithium-Ion Batteries of Electric Bicycles," pp. 1–20, 2025.