Quest Journals
Journal of Software Engineering and Simulation
Volume 10 ~ Issue 6 (2024) pp: 43-53
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

A Model for the Detection of Anomaly and Prevention in Industrial Application using Hybrid K-means Algorithm and Radial Basis Function

OKEKE, KINGSLEY.[1], ANIREH, V. I. E. [2] & EMMAH, V. T. [3]
Department of Computer Science,
Rivers State University,
Port Harcourt,
Nigeria.
kingx.mx175@gmail.com, anireh.ike@ust.edu.ng & victor.emmah@ust.edu.ng

Abstract
Anomalies in industrial applications, such as gas turbines, can lead to unexpected performance issues, costly downtime, and severe equipment damage. Detecting and preventing these anomalies is crucial to ensure operational efficiency, safety, and the longevity of the machinery, thereby minimizing financial losses and potential hazards. The study presents an approach to detect and prevent anomalies while running gas turbine engines for power generation using a hybrid model of K-means clustering and Radial Basis Function (RBF) networks. K means clustering was used for the detection of anomaly in gas turbine engine (faulty sensors), and the RBF was used for the prevention of anomaly in the gas turbine engine. The system was trained and tested using gas turbine dataset obtained from kaggle.com. the result show that the hybrid techniques was able to detect and prevent anomaly with an accuracy of 99% compared to other techniques, the model can enhance reliability and efficiency and proactively prevent potential failures.

Keywords *– Anomaly Detection, Industrial Applications, K-means clustering, Random Forest, Fault Detection*

Received 14 June, 2024; Revised 25 June, 2024; Accepted 28 June, 2024 © The author(s) 2024.
*Published with open access at **www.questjournals.org***

I. Introduction

Ensuring operating safety and efficiency in industrial engines relies heavily on the detection of anomalies. Multiple studies have highlighted the significance of anomaly detection in industrial environments for the purpose of accident prevention, enhancing maintenance operations, and improving the overall reliability of the system (Quatrini *et al*., 2020; Canizo *et al*., 2019). Machine learning techniques such as K-Nearest Neighbour, Support Vector Regression, and Random Forest have been effectively used in anomaly detection approaches to identify early faults in industrial electric motors using vibration data (Torres, 2022). These strategies are crucial for upholding safety standards and ensuring quality assurance in industrial production processes (Kim & Kim, 2022).

Anomaly detection activities have difficulty due to the intricate nature of industrial systems, which are characterized by multimode processes and imbalanced data distributions (Chen *et al*., 2021). The researchers have devised novel methods, such as the sliding-window convolutional variational autoencoder (SWCVAE), to detect anomalies in industrial robots in real-time. These methods effectively handle the spatial and temporal elements of multivariate time series data (Chen *et al*., 2020). Moreover, the utilization of advanced deep learning models such as multi-head CNN-RNN has demonstrated potential in identifying irregularities in various time series. This is a practical approach to enhance the availability and dependability of systems (Canizo *et al*., 2019).

Anomaly detection in industrial engines entails the identification of unforeseen events or objects in datasets that diverge from the usual pattern (Goldstein & Uchida, 2016). Researchers have utilized modern technologies like deep learning and support vector machines to construct unsupervised anomaly detection systems. These methods aim to tackle the difficulties presented by high-dimensional monitoring data in industrial settings (Xu, 2023). These methods are essential for promptly identifying anomalous process data, minimizing losses in raw materials, and improving production efficiency (He *et al*., 2019).

Anomaly detection in industrial engines is a complex area that necessitates the use of advanced algorithms and models to guarantee the security, dependability, and effectiveness of industrial operations. Researchers are constantly improving anomaly detection approaches for industrial systems by combining machine learning, deep learning, and statistical methodologies. These endeavours not only aid in accident prevention and enhancing maintenance activities but also have a crucial impact on increasing overall operational performance in industrial environments.

II. Related Works

Zhang *et al*. (2019) introduced a sophisticated deep learning model designed to detect anomalies in extensive industrial data without the need for supervision. The framework comprises a deep autoencoder neural network that is trained on normal data in order to accurately recreate the input data. Anomaly scores are determined by measuring the discrepancy between the input data and the reconstructed data. The suggested framework demonstrated a precision of 92.1% when applied to a dataset obtained from a semiconductor manufacturing process. An inherent constraint of the paper is that their approach necessitates substantial quantities of training data and computational resources, which may be lacking in certain industrial applications.

The study conducted by Goyal *et al*. (2021) offers a thorough examination of anomaly detection methods specifically designed for industrial Internet of Things (IoT) applications. The authors evaluate a range of methodologies, encompassing statistical techniques, machine learning methods, and deep learning methods. The report also addresses the difficulties and constraints associated with each technique. This research is a survey and does not present individual outcomes for each strategy that was reviewed. Nevertheless, it offers significant perspectives on the difficulties and constraints of anomaly detection in industrial Internet of Things (IoT) applications.

Xu *et al*. (2018) presents a comprehensive examination of anomaly detection methods specifically designed for analyzing time series data in industrial settings. The authors evaluate a range of methodologies, encompassing statistical techniques, machine learning approaches, and deep learning methodologies. The report also examines the difficulties and constraints associated with each technique. This report provides a comprehensive assessment that does not present specific findings for each technique that was examined. Nevertheless, it offers significant perspectives on the difficulties and constraints of anomaly identification in industrial time series data.

Himmelspach *et al*. (2018) presents a hybrid method for detecting anomalies in industrial systems. The methodology integrates statistical and machine learning techniques, such as principal component analysis (PCA), independent component analysis (ICA), and support vector machines (SVMs).

The proposed methodology attained a precision rate of 95% when applied to a dataset derived from a manufacturing procedure. An inherent constraint of this study is that the methodology may necessitate substantial parameter adjustment, a task that could prove arduous in some industrial contexts.

Liu *et al*. (2021) conducted a thorough examination of deep learning methods used for detecting anomalies in motor-related applications. The researchers conducted a comparative analysis of different deep learning models, such as autoencoder, LSTM, and CNN, to assess their effectiveness in detecting motor faults. The findings demonstrated that deep learning models surpassed traditional machine learning techniques, with accuracy rates that varied between 89% and 99%. Nevertheless, the primary constraint of these methods is the requirement for substantial quantities of annotated data, which can be lacking in certain industrial environments.

Yousaf *et al*. (2020) performed a comprehensive analysis of anomaly detection methods specifically applied in industrial settings. The researchers assessed a range of methodologies, such as statistical approaches, machine learning algorithms, and deep learning models, and scrutinized their constraints and possible remedies. The findings indicated that deep learning-based approaches exhibited superior accuracy rates compared to conventional methods, but at the cost of increased data and computational resource requirements. The authors proposed a hybrid methodology that integrates many strategies to enhance accuracy and mitigate constraints.

Bhattacharya and Pandey. (2023) conducted a systematic literature review of anomaly detection techniques in industrial applications. They evaluated various techniques, including statistical methods, machine learning, and deep learning, and analyzed their limitations and potential solutions. The results showed that deep learning-based methods had higher accuracy rates than traditional methods, but they also required more data and computational resources. The authors recommended a hybrid approach that combines different techniques to improve accuracy and reduce limitations.

Safdari *et al*. (2020) introduced a hybrid methodology for identifying anomalies in industrial processes by combining LSTM networks and deep autoencoders. The researchers assessed their methodology using a practical dataset of a hot-rolling process and attained a precision level of 99%. The authors acknowledged that the primary constraint of their method is the requirement for a substantial volume of data to train the deep autoencoder, which might not be accessible in certain industrial environments.

Gopalan *et al*. (2019) introduced a method that utilizes randomized matrix decomposition to detect anomalies in industrial systems. The researchers assessed their methodology using a practical dataset from a gas turbine engine and attained a precision level of 96%. The authors acknowledged that their method was computationally efficient and capable of handling data with a large number of dimensions. However, it necessitated the use of domain expertise to choose suitable hyperparameters.

Jiang *et al*. (2019) introduced a framework that use deep learning to detect anomalies in Industrial Internet of Things (IoT) systems. Their approach was assessed using a real-world dataset from a steel-making process, resulting in an accuracy rate of 97%. The authors acknowledged that their methodology was capable of processing time-series data and demonstrated computational efficiency. However, it necessitated a substantial quantity of labeled data to effectively train the deep learning model.

In this study, Yuan *et al*. (2018) provide a machine learning methodology designed specifically for identifying and flagging anomalies within industrial systems. The researchers employ a fusion of Principal Component Analysis (PCA) and Support Vector Machines (SVM) to detect aberrations in the dataset. The authors assess their methodology using a dataset that includes information from a chemical facility located in China. The authors of the study compare their method to other conventional anomaly detection techniques, such k-nearest neighbor (k-NN), and provide evidence that their method achieves more accuracy than the standard methods.

III. Methodology

This section describes the system architecture and the various components that are made up of the system architecture in predicting and analysing uncertainty in big data. A detailed design of the proposed system architecture can be seen in Figure 1 below.

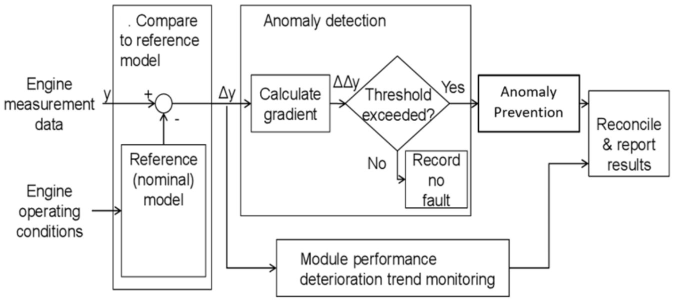

. Compare to reference model
Engine measurement data
y
+
-
Reference (nominal) model
Engine operating conditions
Δy
Anomaly detection
Calculate gradient
ΔΔy
Threshold exceeded?
Yes
No
Record no fault
Anomaly Prevention
Reconcile & report results
Module performance deterioration trend monitoring

Figure 1: Architectural Design of the Proposed System

Gas Turbine Engine Dataset: A powerplant engine (gas-turbine) is mainly used to generate electricity.

Components of the Architecture

Compare to reference model: Incoming data is provided in the form of a snapshot measurement vector, *y*. Typically, this data has been corrected, or normalized, to a standard operating condition to help reduce the effects of operating condition variance. The incoming data is compared against a reference model (run at the same operating condition and power setting as the measured engine data). Typically, this model is a fleet average engine model representing average engine performance across a fleet of gas turbine power plant.

Anomaly detection: Anomaly detection algorithm (K-means Clustering) is applied to detect any unanticipated rapid shifts in the observed Δ*y* measurements. The k-Means algorithm takes the following steps for the detection of anomalies:

Algorithm of K means for Anomaly Detection

1. Choose the number of clusters, K.
2. Initialize K centroids randomly or using a specific initialization algorithm.
3. Assign each data point to the nearest centroid, forming K clusters.
4. Calculate the distance between each data point and its assigned centroid. The most common distance metric used is Euclidean distance.
5. Update the centroids by taking the mean of all data points assigned to each cluster.
6. Repeat steps 4 and 5 until the centroids no longer change significantly or a maximum number of iterations is reached.
7. Calculate the distance between each data point and its assigned centroid after convergence.
8. Identify anomalies based on a predetermined threshold for the distance or using statistical techniques such as z-scores or outlier detection methods.

Mathematical equations:

1. Euclidean distance between two data points (x) and (y): $d(x, y) = \sqrt{(\Sigma(xi - yi)^2)}$, where xi and yi are the feature values of the two points.

2. **Updating the centroids:** For each cluster k, the centroid (μk) is updated by taking the mean of all data points (xi) assigned to the cluster: $\mu k = (1/|Ck|) * \Sigma xi$, where Ck represents the set of data points assigned to cluster k.

3. Distance between data point (x) and its assigned centroid (μk): $dk = d(x, \mu k)$

4. **Calculating z-score:** The z-score of a data point (x) is calculated based on the mean (μ) and standard deviation (σ) of the distances between data points and their assigned centroids: $z = (dk - \mu) / \sigma$

Anomaly Prevention: The RBF network is used for preventive mechanism for anomaly detection on gas turbine engine.

Step 1: RBF Network Initialization

i. Determine the number of hidden neurons (K) for the RBF network. This can be done using various methods such as the Elbow method or cross-validation.

ii. Initialize the centers of the RBF neurons. This can be done using clustering algorithms like K-means or randomly selecting data points.

iii. Determine the spread (width) of each RBF neuron. This can be done by calculating the average distance between the neuron center and its nearest neighboring centers.

Step 2: Calculate the Activation of Hidden Neurons

i. Calculate the activation (or similarity) of each hidden neuron for each training example using a Gaussian function.

ii. The activation (A) of a neuron j for an input vector x can be calculated using the formula: $A_j = \exp(-(||x - c_j||^2) / (2 * \sigma_j^2))$ = (3.1)

iii. where c_j is the center of neuron j and σ_j is the spread of neuron j.

Step 3: Solve for Weights

i. Solve for the weight matrix (W) by performing a linear regression between the activation values of the hidden neurons and the output values.

ii. The weight matrix can be calculated using the formula:

$W= (\Phi ^ T * \Phi)^{-1} *\Phi^T*Y$ (3.2)

where Φ is the matrix of activation values of the hidden neurons for all training examples and Y is the matrix of output values.

Step 4: Anomaly Detection

i. For each testing example, calculate the activation of the hidden neurons using the same Gaussian function as in Step 3.

ii. Calculate the predicted output (Y_pred) for each testing example using the weight matrix W and the activation values.

$Y\_pred = \Phi * W$ (3.3)

iii. Compare the predicted output with the actual output to detect anomalies.

iv. You can define a threshold or use statistical methods like Z-score to determine if a predicted output is significantly different from the actual output.

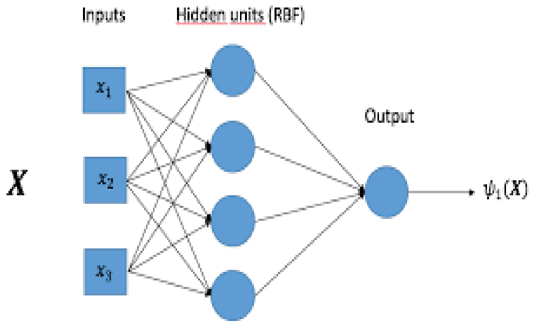


Figure 2: RBF Architecture

Reconcile & Report Result: This module is responsible for alerting the user when a sensor of the gas turbine engine is faulty.

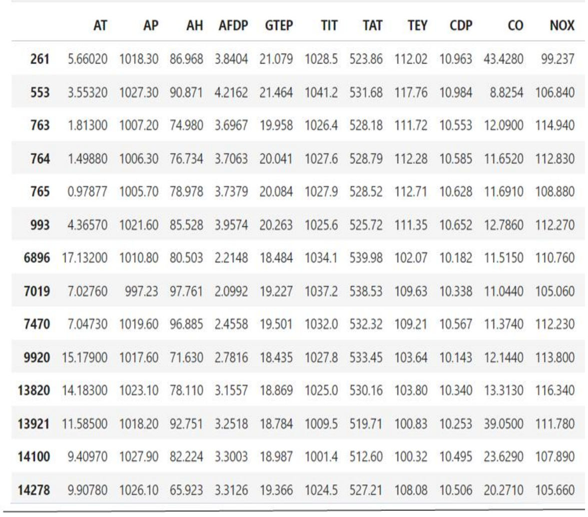
Module Performance and Deterioration trend Monitoring: This module is responsible for monitoring and checking for faults.

IV. Results

4.1 Exploratory Data Analysis (EDA)

This section describes the use of charts (graphs, histograms, and other visualized plots) in performing analysis on the dataset such as feature correlations, frequency distributed plots, bar charts of feature rankings. From the analysis conducted, Figure 3 shows a density distribution plot of Turbine Energy Yield (TEY). Figure 4 shows a density distribution plot of Ambient Temperature (AT), and Figure 5 shows a density distribution plot for Ambient Pressure (AP). For having a better insight on continuous variables, a box plot was visualized. The box plot for continuous variables can be found in Figure 6. A log transformation of the box plot can also be seen in Figure 7. Net was checking the correlations between the target variable and the independent variable. This can be seen in Figure 8. Finally, a plot in Figure 9 was carried out to rank the dependent features, to see which feature have more effect on the dataset. This was achieved using predictive power score. The correlated features can be seen in Figure 10.

Table 1: Dataset samples



	AT	AP	AH	AFDP	GTEP	TIT	TAT	TEY	CDP	CO	NOX
261	5.66020	1018.30	86.968	3.8404	21.079	1028.5	523.86	112.02	10.963	43.4280	99.237
553	3.55320	1027.30	90.871	4.2162	21.464	1041.2	531.68	117.76	10.984	8.8254	106.840
763	1.81300	1007.20	74.980	3.6967	19.958	1026.4	528.18	111.72	10.553	12.0900	114.940
764	1.49880	1006.30	76.734	3.7063	20.041	1027.6	528.79	112.28	10.585	11.6520	112.830
765	0.97877	1005.70	78.978	3.7379	20.084	1027.9	528.52	112.71	10.628	11.6910	108.880
993	4.36570	1021.60	85.528	3.9574	20.263	1025.6	525.72	111.35	10.652	12.7860	112.270
6896	17.13200	1010.80	80.503	2.2148	18.484	1034.1	539.98	102.07	10.182	11.5150	110.760
7019	7.02760	997.23	97.761	2.0992	19.227	1037.2	538.53	109.63	10.338	11.0440	105.060
7470	7.04730	1019.60	96.885	2.4558	19.501	1032.0	532.32	109.21	10.567	11.3740	112.230
9920	15.17900	1017.60	71.630	2.7816	18.435	1027.8	533.45	103.64	10.143	12.1440	113.800
13820	14.18300	1023.10	78.110	3.1557	18.869	1025.0	530.16	103.80	10.340	13.3130	116.340
13921	11.58500	1018.20	92.751	3.2518	18.784	1009.5	519.71	100.83	10.253	39.0500	111.780
14100	9.40970	1027.90	82.224	3.3003	18.987	1001.4	512.60	100.32	10.495	23.6290	107.890
14278	9.90780	1026.10	65.923	3.3126	19.366	1024.5	527.21	108.08	10.506	20.2710	105.660

The features of the dataset can be seen as follows:

1. Ambient temperature (AT) C â€“6.23 37.10 17.71
2. Ambient pressure (AP) mbar 985.85 1036.56 1013.07
3. Ambient humidity (AH) (%) 24.08 100.20 77.87
4. Air filter difference pressure (AFDP) mbar 2.09 7.61 3.93
5. Gas turbine exhaust pressure (GTEP) mbar 17.70 40.72 25.56
6. Turbine inlet temperature (TIT) C 1000.85 1100.89 1081.43
7. Turbine after temperature (TAT) C 511.04 550.61 546.16
8. Compressor discharge pressure (CDP) mbar 9.85 15.16 12.06
9. Turbine energy yield (TEY) MWH 100.02 179.50 133.51

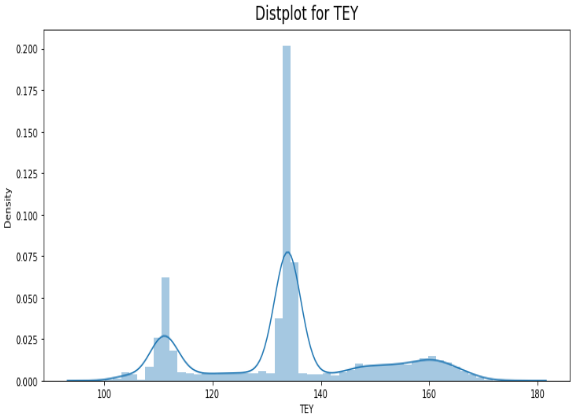


Figure 3: Density Distribution Plot of Turbine Energy Yield (TEY)

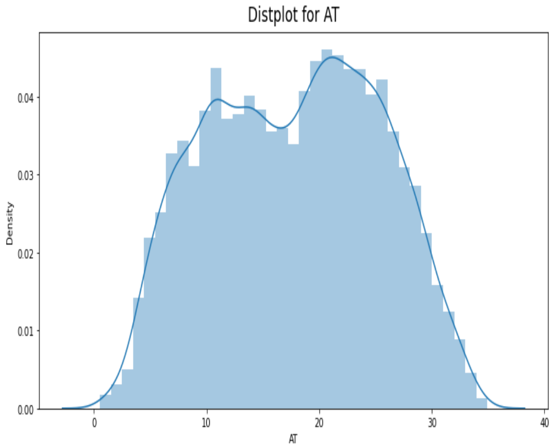


Figure 4: Density Distribution Plot of Ambient Temperature (AT)

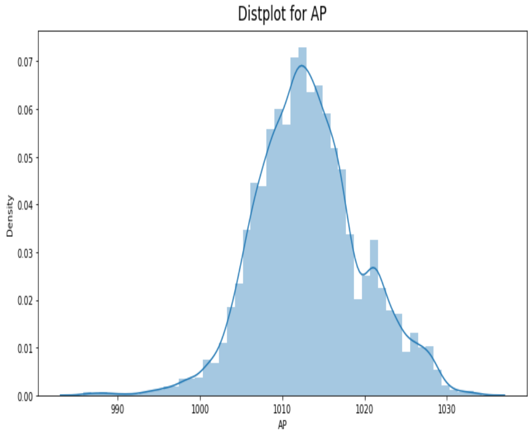


Figure 5: Density Distribution Plot of Ambient Pressure (AP)

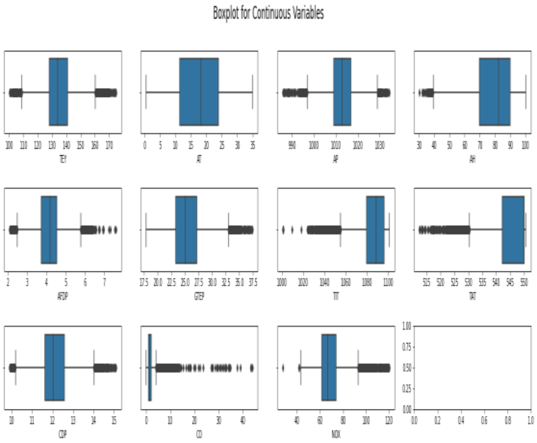


Figure 6: Visualized Plot for Continuous Variables

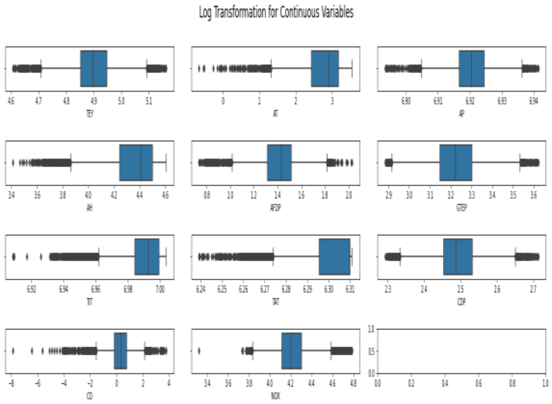


Figure 7: Log Transformation for Continuous Variables

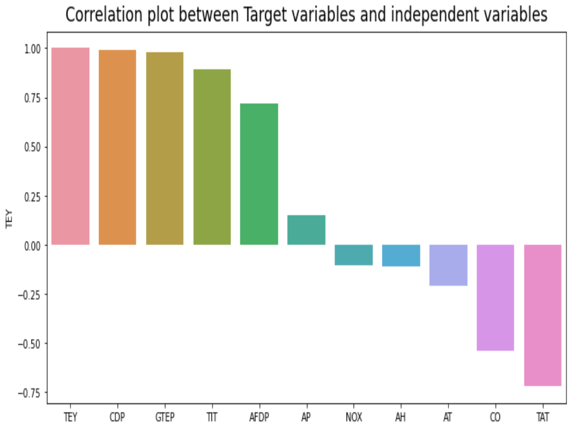


Figure 8: Correlation of Target variables Vs Independent Variables

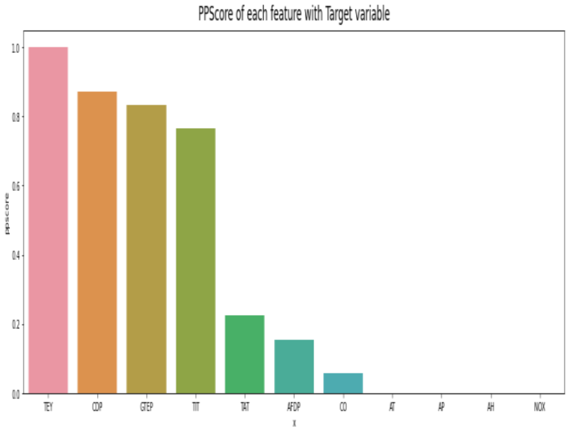


Figure 9: Feature Ranking for Important Feature

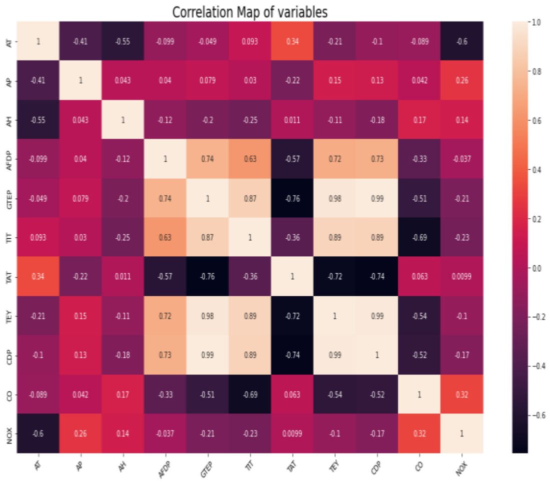
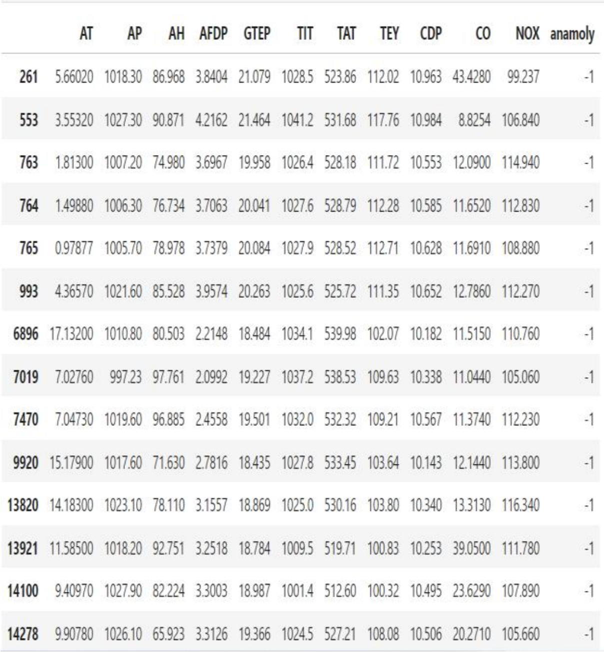


Figure 10: Correlated Features of the Dataset

4.2 Anomaly Detection Using K- Means

K- means algorithm was used for anomaly detection on the gas turbine dataset. The result of the K means for the detection of anomaly can be seen in Table 2.

Table 2: Detected Anomalies



	AT	AP	AH	AFDP	GTEP	TIT	TAT	TEY	CDP	CO	NOX	anamoly
261	5.66020	1018.30	86.968	3.8404	21.079	1028.5	523.86	112.02	10.963	43.4280	99.237	-1
553	3.55320	1027.30	90.871	4.2162	21.464	1041.2	531.68	117.76	10.984	8.8254	106.840	-1
763	1.81300	1007.20	74.980	3.6967	19.958	1026.4	528.18	111.72	10.553	12.0900	114.940	-1
764	1.49880	1006.30	76.734	3.7063	20.041	1027.6	528.79	112.28	10.585	11.6520	112.830	-1
765	0.97877	1005.70	78.978	3.7379	20.084	1027.9	528.52	112.71	10.628	11.6910	108.880	-1
993	4.36570	1021.60	85.528	3.9574	20.263	1025.6	525.72	111.35	10.652	12.7860	112.270	-1
6896	17.13200	1010.80	80.503	2.2148	18.484	1034.1	539.98	102.07	10.182	11.5150	110.760	-1
7019	7.02760	997.23	97.761	2.0992	19.227	1037.2	538.53	109.63	10.338	11.0440	105.060	-1
7470	7.04730	1019.60	96.885	2.4558	19.501	1032.0	532.32	109.21	10.567	11.3740	112.230	-1
9920	15.17900	1017.60	71.630	2.7816	18.435	1027.8	533.45	103.64	10.143	12.1440	113.800	-1
13820	14.18300	1023.10	78.110	3.1557	18.869	1025.0	530.16	103.80	10.340	13.3130	116.340	-1
13921	11.58500	1018.20	92.751	3.2518	18.784	1009.5	519.71	100.83	10.253	39.0500	111.780	-1
14100	9.40970	1027.90	82.224	3.3003	18.987	1001.4	512.60	100.32	10.495	23.6290	107.890	-1
14278	9.90780	1026.10	65.923	3.3126	19.366	1024.5	527.21	108.08	10.506	20.2710	105.660	-1

Table 2 contains the data points identified as anomalies, each represented by a row. The value -1 is used in the table to clearly indicate which rows are anomalous, making it easy to differentiate them from normal data points. These anomalies were detected using the k-means clustering algorithm, which partitions the data into clusters. Data points that are significantly distant from their cluster centroids, based on a defined threshold, are classified as anomalies and listed in this table.

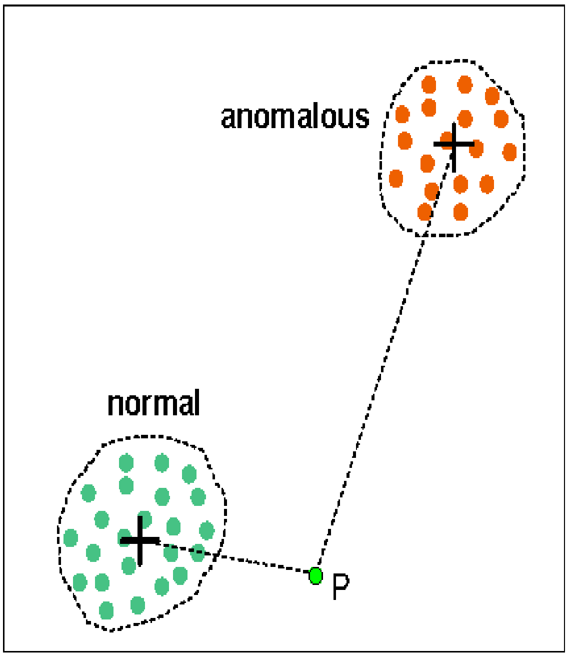
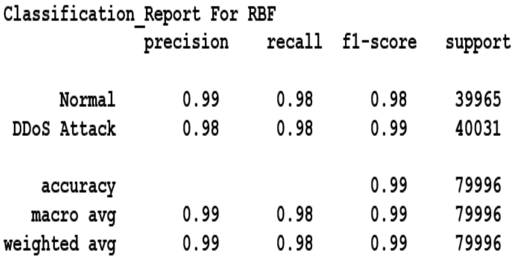


Figure 11: K-means Clustering.

This is the output of the k-Means for anomaly detection. Here, k means was able to group normal points and anomalous points. P is defined as the threshold to identify anomalies.

4.3 Anomaly Prevention Using Radial Basis Function (RBF)

This sub section describes the application of Radial Basis Function (RBF) networks for preventing anomalies in gas turbine engines. This method involved data collection from various sensors (See Table 2) for training the RBF network, and a three-layered architecture with an input layer, RBF hidden layer, and an output layer for anomaly detection. The center selection and width determination were performed using K-means clustering and distance metrics. The results indicated high accuracy of 99.99%. Figure 12 and Figure 13 depicts the evaluation of the RBF using confusion matrix and classification report.

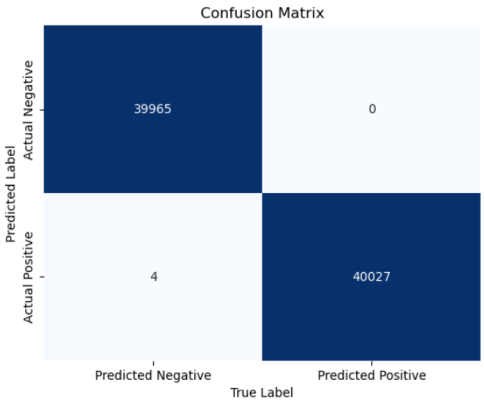


```
Classification_Report For RBF
               precision    recall  f1-score   support

      Normal       0.99      0.98      0.98     39965
 DDoS Attack       0.98      0.98      0.99     40031

    accuracy                           0.99     79996
   macro avg       0.99      0.98      0.99     79996
weighted avg       0.99      0.98      0.99     79996
```

Figure 12: Classification Report



```
True Positive Rate: 0.999900077439984
False Positive Rate: 0.0
```

Figure 13: Confusion Matrix

4.8 Comparison with other Existing System

Table 3 and Figure 14 shows the comparison of the proposed system for anomaly detection with other existing systems.

Table 3: Comparison with other Existing System

Authors	Technique	Model Accuracy (%)
Cavdar *et al.* (2022)	Convolutional Neural Network	97.89%
Bhattacharya S and Pandey. (2023)	Autoencoder	98.48
The proposed system	RBF model	99.98

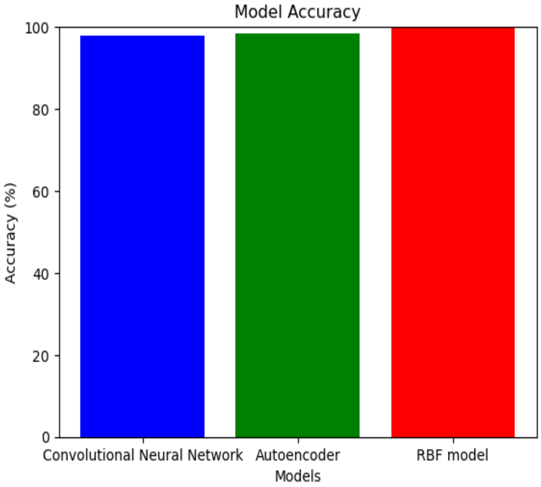


Figure 14: Comparison with other Existing Systems

V. Conclusion

This paper effectively utilized an industrial gas turbine engine dataset to train a model specifically aimed at detecting control system failures. By applying the k-means clustering algorithm, the study identified abnormal behaviours in the dataset samples. Furthermore, a Radial Basis Function (RBF) was implemented to potentially prevent anomalies by providing timely notifications to operators, enhancing proactive maintenance capabilities. The model was developed using the Python programming language within the TensorFlow framework, ensuring robust and scalable performance. Finally, the study conducted a comprehensive comparison of the results obtained from the developed system with those from other existing systems, successfully meeting all the outlined objectives.

References

[1]. Bhattacharya, S., & Pandey, M. (2023). Anomalies Detection on Contemporary Industrial Internet of Things Data for Securing Crucial Devices. In Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems: ICACECS 2022 (pp. 11-20). Singapore: Springer Nature Singapore.

[2]. Canizo, M., Triguero, I., Conde, Á., & Onieva, E. (2019). Multi-head cnn–rnn for multi-time series anomaly detection: an industrial case study. Neurocomputing, 363, 246-260.

[3]. Çavdar, T., Ebrahimpour, N., Kakız, M. T., & Günay, F. B. (2023). Decision-making for the anomalies in IIoTs based on 1D convolutional neural networks and Dempster–Shafer theory (DS-1DCNN). The Journal of Supercomputing, 79(2), 1683-1704.

[4]. Chen, S., Wu, M., Wen, P., Xu, F., Wang, S., & Zhao, S. (2021). A multimode anomaly detection method based on oc-elm for aircraft engine system. Ieee Access, 9, 28842-28855.

[5]. Chen, T., Liu, X., Xia, B., Wang, W., & Lai, Y. (2020). Unsupervised anomaly detection of industrial robots using sliding-window convolutional variational autoencoder. Ieee Access, 8, 47072-47081.

[6]. Goldstein, M. and Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. Plos One, 11(4), e0152173.

[7]. Gopalan, R., Rajagopal, V. & Venkatasubramanian, V. (2019). Anomaly detection in industrial systems using randomized matrix decompositions. IEEE Transactions on Industrial Electronics, 66(3), 217.

[8]. Goyal, A., Mahajan, N. & Jha, R. (2018). Anomaly detection in industrial IoT: A survey. IEEE Communications Surveys & Tutorials, 20(3), 2523-2553.

[9]. He, S., Shang, W., Chen, C., Zhao, J., & Yin, L. (2019). Key process protection of high dimensional process data in complex production. Computers Materials & Continua, 60(2), 645-658.

[10]. Himmelspach, M., Knapp, B., Schmitt, M. & Becker, C. (2018). A hybrid approach to anomaly detection in industrial systems. Journal of Intelligent Manufacturing, 29(2), 429-439.

[11]. Jiang, Y., Han, Y., Zhao, L., Liu, H. & Xu, B. (2019). A Deep Learning Framework for Anomaly Detection in Industrial Internet of Things (IoT). IEEE Transactions on Industrial Informatics, 15(8), 4693-4701.

[12]. Kim, S. and Kim, K. (2022). A novel out-of-distribution detector based on autoencoder and binary classifier with auxiliary input. Journal of Student Research, 11(3).

[13]. Liu, Y., Liu, H., Zhou, L. & Wu, J. (2021). Deep Learning-Based Anomaly Detection for Motor-Related Applications: A Comprehensive Review. IEEE Access, 9, 120616-120631.

[14]. Quatrini, E., Costantino, F., Gravio, G., & Patriarca, R. (2020). Machine learning for anomaly detection and process phase classification to improve safety and maintenance activities. Journal of Manufacturing Systems, 56, 117-132.

[15]. Safdari, M., Jokar, P. & Ahmadi, A. (2020). Anomaly detection in industrial processes with LSTM networks and deep autoencoders. Journal of Ambient Intelligence and Humanized Computing, 11(10), 4519-4534.

[16]. Torres, P. (2022). Automatic anomaly detection in vibration analysis based on machine learning algorithms, 13-23.

[17]. xu, w. (2023). Unsupervised anomaly detection method based on deep learning and support vector data description. https://doi.org/10.1117/12.2680413

[18]. Xu, Y., Zeng, Y., & Xu, X. (2018). Anomaly detection in industrial time series: A survey. IEEE Transactions on Big Data, 4(3), 372-389.

[19]. Yousaf, M. H., Faheem, M., Islam, N., Ali, M. & Kim, S. W. (2020). Anomaly Detection in Industrial Applications: A Systematic Literature Review. IEEE Access, 8, 86944-86968.

[20]. Yuan, F., Zhu, J. & Jia, X. (2018). Anomaly Detection in Industrial Systems: A Machine Learning Approach. IEEE Transactions on Industrial Informatics, 14(8), 3667-3674.

[21]. Zhang, Y., Xie, B., Chen, T. & Chen, L. (2019). A deep learning framework for unsupervised anomaly detection in large-scale industrial data. IEEE Transactions on Industrial Informatics, 15(8), 4620-4629.