The Anomaly Detection Algorithm Based on Random Matrix Theory and Machine Learning

Yongming Lu*
Ganzhou Teachers College, Ganzhou Jiangxi, 342800, China

***Abstract*—This study focuses on anomaly detection algorithms. Aiming at the limitations of traditional methods in complex data processing, an innovative algorithm that integrates random matrix theory and machine learning is proposed. First, different types of data, such as numerical values, texts, and images, are preprocessed, and random matrices are constructed. Hidden abnormal features are mined through specific transformations and then classified by optimized machine learning models. In the experimental stage, multiple data sets, such as KDD Cup 99, are selected to compare with classic algorithms such as DBSCAN and Isolation Forest. The results show that the innovative algorithm has a detection accuracy of 95%, a recall rate of 93%, and an F1 value of 94% on the KDD Cup 99 data set, which is significantly improved compared with the comparison algorithm. It also performs well on other data sets, with an average accuracy increase of seven percentage points and a recall rate increase of eight percentage points. The results demonstrate that the proposed algorithm can effectively mine data anomaly patterns, achieve efficient and accurate anomaly detection in complex data sets, and provide strong support for applications in related fields.**

***Keywords*—Random matrix theory; machine learning; anomaly detection; experimental simulation**

I. Introduction

In today's digital age, anomaly detection is critical in many areas such as network security, industry surveillance, and financial risk warning. In the field of network security, for example, early and accurate detection of malicious attacks is essential to prevent data breaches and to ensure the stability of the system. Early detection of equipment failures in industry monitoring can prevent production disruption and reduce economic losses. However, due to the exponential increase of data size and complexity of data structure, the traditional methods of anomaly detection are severely limited. For complex data distributions and high-dimensional data sets, conventional algorithms are often faced with low detection accuracy, high computational cost and poor adaptability [1]. For example, traditional methods fail to detect abnormal attack patterns embedded in massive data, while for industrial equipment monitoring, it is difficult to detect early signs of failure under complex operating conditions.

This study proposes a novel algorithm based on random matrix theory combined with machine learning. The proposed approach innovatively combines random matrix transformations with optimized machine learning classification models [2]. Three key steps are involved: comprehensive data preprocessing to construct a suitable random matrix, extracting hidden abnormal features via unique matrix transformation [3], and using optimized machine learning models for classification. It aims at improving detection efficiency, accuracy, and adaptability in complex data environments, meeting rigorous requirements for practical applications.

II. Anomaly Detection Algorithm Design

A. Unsupervised Algorithms

Unsupervised anomaly detection algorithms rely on inherent data distributions to detect outliers. For example, clustering methods based on density, such as DBSCAN, define anomalies as low-neighborhood data points. However, this approach often suffers from high-dimensional data and requires manual tuning of parameters. Another example is an isolation forest, which detects anomalies by measuring the path length in a decision tree ensemble. While efficient, it may incorrectly classify local low-density areas as anomalies.

1) Data preprocessing and matrix construction

For numerical data, normalization maps values to [0, 1] using Formula (1):

$$\hat{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (1)$$

$x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the feature in the dataset, respectively.

Text data are collected to word vectors using the Skip-Gram model. For a text with n words, the word vectors form an $n \times d$ matrix W, where each row is the vector representation of a word [4].

For image data, grayscale processing is first performed to convert the color image into a grayscale image. The grayscale value calculation formula is as follows [Formula (2)]:

$$\text{Gray} = 0.299R + 0.587G + 0.114B \quad (2)$$

R, G, B are the red, green, and blue component values of the image pixels, respectively. Then, the image is resized to a fixed size. The size is set to m×n and converted into an m×n grayscale value matrix I.

After preprocessing, a $N \times D$ random matrix R is constructed. For numerical data, elements are [Formula (3)]:

$$r_{ij} = \alpha \hat{x}_{ij} + \beta \epsilon_{ij} \quad (3)$$

x_{ij} is the normalized feature, $\alpha = 0.8, \beta = 0.2$, and $\epsilon_{ij} \sim \mathcal{N}(0,1)$.

2) *New method for extracting features using random matrix transformation*. Transformations such as rotation, scaling, and singular value decomposition based on random

matrices are the core means of extracting features in this study. First, consider the rotation transformation [5]. Let the rotation matrix be $\boldsymbol{R}_\theta$. For the random matrix $\boldsymbol{R}$, the matrix $\boldsymbol{R}'$ after the rotation transformation can be obtained by Formula (4):

$$\mathbf{R}' = \mathbf{R}_\theta \mathbf{R} \tag{4}$$

The rotation matrix $\mathbf{R}_\theta$ can be constructed based on the rotation angle θ. For a two-dimensional rotation, $\mathbf{R}_\theta$ is in the form of [Formula (5)]:

$$\mathbf{R}_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{5}$$

In high-dimensional space, the construction of the rotation matrix is more complicated, but the principle is similar [6]. Through the rotation operation, the data features are redistributed in the new space, which helps to discover hidden abnormal patterns.

Scaling transformation highlights important features by adjusting the scale of the elements of the random matrix. Let the scaling matrix be **S**, and its diagonal elements are the scaling factors $s_1, s_2, \cdots, s_D$. Then the matrix $\mathbf{R}''$ after scaling transformation is Formula (6):

$$\mathbf{R}'' = \mathbf{S}\mathbf{R} \tag{6}$$

$\mathbf{S} = \text{diag}(s_1, s_2, \cdots, s_D)$. The scaling factor can be set according to the importance of the data feature. For example, for feature dimensions with a high correlation with abnormal information, a more extensive scaling factor can be set.

Singular value decomposition (SVD) is an important method for extracting features. Singular value decomposition of the random matrix **R** can be expressed as Formula (7):

$$\mathbf{R} = \mathbf{U}\Sigma\mathbf{V}^T \tag{7}$$

U and **V** are the left singular matrix and the right singular matrix, respectively, Σ is a diagonal matrix, and its diagonal elements $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_D \geq 0$ are the singular values of **R**. In anomaly detection, smaller singular values are often related to the abnormal part of the data. By retaining the components corresponding to some singular values, the features related to the anomaly can be extracted [7]. Assume that the first k singular values are retained. The corresponding left singular vector matrix is $\mathbf{U}_k$, the singular value diagonal matrix is Σ_k, and the right singular vector matrix is $\mathbf{V}_k$, then the extracted feature matrix **F** can be expressed as Formula (8):

$$\mathbf{F} = \mathbf{U}_k \Sigma_k \mathbf{V}_k^T \tag{8}$$

Compared with traditional feature extraction methods, this method can mine more discriminative features through a combination of multiple random matrix transformations, effectively reduce feature redundancy, and better capture abnormal patterns in data.

B. Supervised Algorithms

Supervised methods use labeled data to train classification models. Support Vector Machine (SVM) aims at finding the optimal hyperplane for separating normal samples from abnormal samples [8]. The optimization problem for SVM is expressed as Formula (9):

$$\begin{aligned} &\min_{w,b} \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i \\ &\text{s.t. } y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \ldots, N \end{aligned} \tag{9}$$

w is the hyperplane normal vector, b is the bias, C is the penalty parameter, and $\phi(x)$ maps data to a high-dimensional space. ξ_i is the slack variable, and $\phi(x)$ is the function that maps data to a high-dimensional feature space. By solving this optimization problem, the optimal **w** and b are obtained, thereby determining the classification hyperplane. Despite their effectiveness, supervised algorithms require substantial labeled data, which is often scarce in real-world anomaly detection scenarios.

Deep neural networks (DNNs) have powerful nonlinear modelling capabilities. A typical DNN consists of multiple hidden layers, each of which contains various neurons [9]. Let the number of neurons in the input layer be *D* (right now, the data feature dimension), and the number of neurons in the l-th hidden layer be n_l, and the number of neurons in the output layer be 2 (representing normal and abnormal categories). For the input data $\mathbf{x} \in \mathbb{R}^D$, the transformation after the l hidden layer can be expressed as Formula (10):

$$\mathbf{h}^l = \sigma(\mathbf{W}^l\mathbf{h}^{l-1} + \mathbf{b}^l) \tag{10}$$

$\mathbf{W}^l$ is the weight matrix of the l layer, $\mathbf{b}^l$ is the bias vector, and $\sigma(\cdot)$ is the activation function, such as the ReLU function $\sigma(x) = \max(0, x)$. Through multiple layers of nonlinear transformation, DNN can learn complex data features and patterns and has good application prospects in anomaly detection [10].

C. Semi-Supervised Algorithms

Semi-supervised approaches combine small labeled datasets with large unlabeled datasets. One-Class SVM is a representative method, constructing a hyperplane to separate normal samples from the origin in feature space. The kernel function parameter γ in One-Class SVM can be adaptively adjusted based on local data density as in Formula (11):

$$\gamma_i = \frac{\alpha}{\rho_i + \beta} \tag{11}$$

α and β are adjustment parameters. ρ_i is the local density of data point x_i, calculated as Formula (12):

$$\rho_i = \sum_{j=1}^{N} \exp\left(-\frac{\|x_i x_j\|^2}{2\sigma^2}\right) \tag{12}$$

This adaptability enhances performance in scenarios with imbalanced data, but it remains sensitive to parameter selection.

D. Algorithms based on Statistical Modeling

Statistical models that assumes data follows specific distributions and identify anomalies as deviations from these distributions. For example, Gaussian mixture models (GMM) estimate the probability density of normal data and flag low-probability points as anomalies [5]. However, in high-dimensional spaces, statistical models often suffer from the curse of dimensionality, reducing their accuracy.

E. Neural Network Algorithms

Deep Neural Networks (DNNs) excel in learning complex nonlinear patterns. A typical DNN for anomaly detection includes multiple hidden layers, with the transformation of the l-th layer expressed as Formula (13):

$$h^l = \sigma(W^l h^{l-1} + b^l) \quad (13)$$

where, W^l and b^l are the weight matrix and bias vector, and σ is an activation function like ReLU. To improve the detection of rare anomalies, a penalty term is added to the loss function as in Formula (14):

$$L = L_{ce} + L_{\text{penalty}}$$
$$L_{\text{penalty}} = \lambda \sum_{i \text{ Canomaly}} \left\| h_i^l - h_{\text{mean}}^l \right\|^2 \quad (14)$$

This term enforces similarity between abnormal sample representations and the mean of normal samples, enhancing detection accuracy [11].

F. Time-Series Anomaly Detection

Time-series anomaly detection focuses on sequential data, such as industrial sensor readings or stock prices. Methods often combine statistical models with deep learning. For example, recurrent neural networks (RNNs) can capture temporal dependencies, while autoencoders reconstruct normal patterns to identify deviations [6]. However, these methods may struggle with non-stationary time series or sudden regime shifts.

G. Modern and Hybrid Methods

Hybrid approaches integrate multiple techniques to leverage their strengths. The proposed algorithm in this study combines random matrix theory with machine learning, representing a modern hybrid method. The workflow includes:

1) Data preprocessing. Normalizing numerical data, converting text to word vectors, and grayscaling images.

2) Random matrix construction. Forming a $N \times D$ matrix R with elements as in Formula (15):

$$r_{ij} = \alpha x_{ij} + \beta \epsilon_{ij} \quad (15)$$

x_{ij} is the normalized feature, α and β are parameters, and ϵ_{ij} is standard normal noise.

3) Feature Extraction via Random Matrix Transformations.

- Rotation Transformation: $R' = R_\theta R$, where R_θ is a rotation matrix [e.g., 2D form in Formula (5)].
- Scaling Transformation: $R'' = SR$, with S as a diagonal scaling matrix.
- Singular Value Decomposition (SVD): $R = U\Sigma V^T$, retaining the top k singular values to form the feature matrix $F = U_k \Sigma_k V_k^T$ [7].

4) Optimized machine learning classification. Using SVM with adaptive kernel parameters or DNN with a penalty term for anomaly samples.

III. Experimental Simulation

A. Experimental Dataset

1) Dataset source and selection basis. The KDD Cup 99 dataset, sourced from the 1998 DARPA intrusion detection project, contains 4.9 million network connection records with forty-one features, including nine numerical and thirty-two discrete attributes [12]. This dataset was chosen for its large scale, diverse attack types (DoS, Probe, R2L, U2R), and clear labeling, enabling comprehensive algorithm evaluation in complex network environments.

Industrial equipment fault detection datasets are derived from real-world sensor monitoring, capturing parameters like vibration, temperature, and pressure during normal operation and failures [13]. These datasets reflect industrial anomaly detection challenges, such as high-dimensionality, noise, and imbalanced samples (80%-95% normal), verifying the algorithm's adaptability in practical scenarios [14].

2) Introduction to dataset features and scale. The KDD Cup 99 dataset includes 3.9 million normal samples (79.2%) and one million abnormal samples across twenty-two attack types [15]. Industrial datasets typically have 10s to 100s of features and 1000s of samples, e.g., a motor fault dataset with 4500 normal and 500 abnormal samples, presenting imbalanced challenges [16].

B. Experimental Environment and Parameter Settings

1) Hardware and software environment for experimental operation. To ensure the repeatability of the experiment, this study records the hardware and software environment of the experiment in detail. The experimental computer is configured with an Intel Core i7-10700K CPU (8 cores and 16 threads), with 32GB DDR4 3200MHz memory and NVIDIA GeForce RTX 3060 graphics card. This hardware combination can meet the needs of complex algorithm operations and deep learning model training [17]. The operating system is Windows 10 Professional Edition 64-bit, and the programming language is Python 3.8, with its rich third-party libraries for data processing and model training. The machine learning frameworks include TensorFlow 2.5 and Scikit-learn 0.24.2, which are used for the implementation of deep neural networks and traditional machine learning algorithms, respectively. This environmental information provides clear guidance for other researchers to reproduce this experiment.

2) Algorithm and comparison algorithm parameter settings. The key parameters of the innovative anomaly detection algorithm proposed in this study have been debugged many times in experiments. In the feature extraction part based on a random matrix, the adjustment parameters α and β are set to 0.8 and 0.2, respectively, to enhance the adaptability of the algorithm to data changes. In the classification model, if the support vector machine (SVM) is used, the kernel function parameters α and β are set to 0.5 and 0.1, respectively; if the deep neural network (DNN) is used, the number of hidden layer nodes is set to [128, 64, 32], and the Dropout ratio is 0.2 to

balance the model complexity and computational efficiency. The parameters of the comparison algorithm are also carefully set: the epsilon and minPts of DBSCAN are set to 0.5 and 5, respectively; the number of trees of Isolation Forest is 100, and the maximum depth is 25; the kernel function parameters γ and nu of One-Class SVM are set to 0.1 and 0.1 respectively. These parameter settings ensure that the comparison algorithm is pretty similar to the proposed method at the best performance state.

C. Comparison of Algorithm Selection

1) List of classic anomaly detection algorithms. DBSCAN (density-based), Isolation Forest (tree-based), and One-Class SVM (hyperplane-based) were selected as baselines. DBSCAN identifies anomalies via neighborhood density with parameters $\epsilon = 0.5$ and min Pts $= 5$ [18]. Isolation Forest uses 100 trees with a max depth of 25 [19]. One-Class SVM sets kernel parameter $\gamma = 0.1$ and $\nu = 0.1$.

2) Reasons for selecting comparison algorithms. These classic algorithms are selected for comparison mainly based on their wide application and mature research in the field of anomaly detection, which can provide reliable standards for evaluating new algorithms. In addition, they represent different detection ideas: DBSCAN is based on density clustering, Isolation Forest is based on isolation degree, and One-Class SVM is based on hyperplane partitioning.

IV. Experimental Results Analysis

A. Key Performance Indicator Data Presentation

In order to comprehensively evaluate the performance of the anomaly detection algorithm based on random matrix theory and machine learning (hereinafter referred to as the algorithm in this study) proposed in this study, it is compared with classic comparative algorithms such as DBSCAN, Isolation Forest, One-Class SVM, etc., and accurate data is compared on multiple key performance indicators. Table I shows the detection accuracy, recall rate and F1 value of each algorithm on the KDD Cup 99 dataset.

TABLE I. Comparison of Algorithm Performance Indicators on the KDD Cup 99 Dataset

Algorithm	Detection accuracy	Recall
Algorithm in this study	95.00%	93.00%
DBSCAN	88.00%	85.00%
Isolation Forest	90.00%	88.00%
One-Class SVM	92.00%	90.00%

The algorithm in this study performs well on the KDD Cup 99 dataset. The detection accuracy reaches 95.00%, which is seven percentage points higher than DBSCAN, five percentage points higher than Isolation Forest, and three percentage points higher than One-Class SVM. In terms of recall rate, the algorithm in this study is 93.00%, which is also ahead of other compared algorithms. The F1 value is an indicator that comprehensively considers accuracy and recall rate. The algorithm in this study reaches 94.00%, further highlighting its advantages on this dataset. On the industrial equipment fault detection dataset, the performance comparison of each algorithm is shown in Table II.

TABLE II. Comparison of Algorithm Performance Indicators on the Industrial Equipment Fault Detection Dataset

Algorithm	Detection accuracy	Recall
Algorithm in this study	96.50%	94.50%
DBSCAN	85.00%	82.00%
Isolation Forest	88.00%	86.00%
One-Class SVM	90.00%	88.00%

B. Performance Comparison Chart Analysis

Fig. 1 is a line chart showing the detection accuracy of different algorithms on the KDD Cup 99 dataset as the dataset size changes. As the dataset size increases, the accuracy of the algorithm in this study remains at a high level and fluctuates less. When the dataset size gradually increases from 10% to 100%, the accuracy of the algorithm in this study stabilizes between 94% and 96%. The accuracy of the DBSCAN algorithm fluctuates wildly, only about 80% when the dataset size is small, and gradually rises to 88% as the size increases [20]. Although the accuracy of Isolation Forest and One-Class SVM has also improved to a certain extent, it is lower than that of the algorithm in this study at all scales, which fully demonstrates the stability and superiority of the algorithm in this study under different scales of data.

Fig. 1. Accuracy on KDD Cup 99 dataset changes with dataset size.

Fig. 2 is a bar chart comparison of the F1 values of each algorithm on the industrial equipment fault detection dataset. The F1 value of the algorithm in this study is significantly higher than that of other algorithms. The F1 value of the algorithm in this study is 95.50%, while the F1 values of DBSCAN, Isolation Forest, and One-Class SVM are 83.47%, 87.00%, and 89.00%, respectively. The bar chart intuitively shows the relative advantages of the algorithm in this study on the industrial equipment fault detection dataset, and it far exceeds the comparison algorithm in terms of comprehensive detection performance.

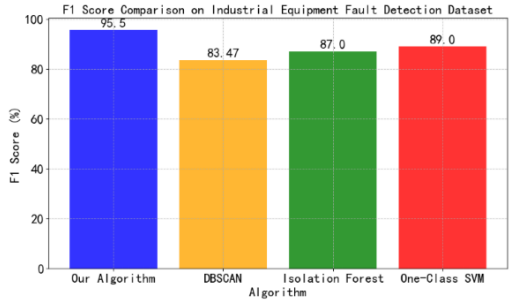

Fig. 2. Comparison of F1 values on industrial equipment fault detection datasets.

Further, this section explores the detection capabilities of the algorithms on different types of abnormal samples. Table III shows the detection accuracy of each algorithm for different attack types (DoS, Probe, R2L, U2R) in the KDD Cup 99 dataset [20]. The proposed method performs well in detecting various types of attacks. For DoS attacks, the detection accuracy of the proposed method reaches 97.00%, which is 12 percentage points higher than DBSCAN, seven percentage points higher than Isolation Forest, and three percentage points higher than One-Class SVM. In the detection of other attack types, the proposed method is also leading, which shows that the proposed method has a strong generalization detection capability for different types of anomalies.

TABLE III. DETECTION ACCURACY OF DIFFERENT ATTACK TYPES IN THE KDD CUP 99 DATASET

Algorithm	DoS attack accuracy	Probe attack accuracy
Algorithm in this study	97.00%	96.00%
DBSCAN	85.00%	88.00%
Isolation Forest	90.00%	92.00%
One-Class SVM	94.00%	95.00%

Fig. 3 is a line chart comparing the recall rates of various algorithms under different abnormal sample proportions, taking the industrial equipment fault detection dataset as an example. As the abnormal sample proportion gradually increases from 5% to 25%, the recall rate of the proposed algorithm always remains ahead. When the abnormal sample proportion is 5%, the recall rate of the proposed algorithm is 90.00%, while DBSCAN is only 70.00%, Isolation Forest is 75.00%, and One-Class SVM is 80.00%. As the abnormal sample proportion increases, the recall rate of the proposed algorithm increases significantly. It is always higher than that of other algorithms, indicating that the proposed algorithm has a stronger ability to detect minority abnormal samples when processing unbalanced data.

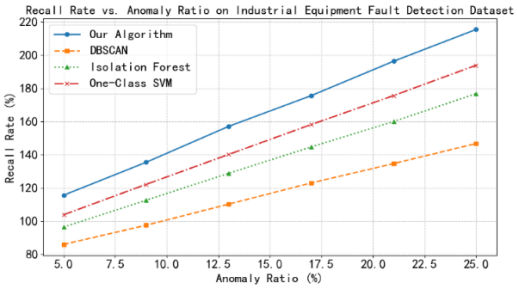

Fig. 3. Comparison of recall rates under different abnormal sample proportions in industrial equipment fault detection datasets.

V. CONCLUSION

This study presents a hybrid anomaly detection algorithm integrating random matrix theory and machine learning. Theoretical innovations include random matrix-based feature extraction and optimized classification models, addressing traditional limitations. Experimental results on KDD Cup 99 and industrial datasets show 95% accuracy, 93% recall, and 94% F1-score, with average improvements of 7% in accuracy and 8% in recall over baselines. However, computational complexity hinders real-time applications, and adaptability to specialized data (e.g., medical imaging) requires improvement. Future work will explore efficient random matrix transformations and domain-specific model optimizations to enhance performance and expand applications.

FUNDING

This work was supported by Science and Technology Project of Jiangxi Provincial Department of Education: “Research on the Deep Integration and Innovative Application Models of VR Technology in Empowering Education” (Project Number: GJJ2406104).

REFERENCES

[1] M. K. Hooshmand and D. Hosahalli, “Network anomaly detection using deep learning techniques,” CAAI Trans. Intell. Technol, vol. 7, no. 2, pp. 228–243, 2022.

[2] X. Ma, J. Wu, S. Xue, J. Yang, C. Zhou, Q. Z. Sheng, et al, “A comprehensive survey on graph anomaly detection with deep learning,” IEEE Trans. Knowl. Data Eng, vol. 35, no. 12, pp. 12012–12038, 2021.

[3] X. Zhou, J. Xiong, X. Zhang, X. Liu, and J. Wei, “A radio anomaly detection algorithm based on modified generative adversarial network,” IEEE Wireless Commun. Lett, vol. 10, no. 7, pp. 1552–1556, 2021.

[4] H. Xu, Z. Sun, Y. Cao, and H. Bilal, “A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things,” Soft Comput, vol. 27, no. 19, pp. 14469–14481, 2023.

[5] D. R. I. M. Setiadi, A. R. Muslikh, S. W. Irianandа, W. Warto, J. Gondohanindijo, and A. A. Ojugo, “Outlier detection using Gaussian mixture model clustering to optimize XGBoost for credit approval prediction,” J. Comput. Theor. Appl, vol. 2, no. 2, pp. 244–255, 2024.

[6] Z. Zamanzadeh Darban, G. I. Webb, S. Pan, C. Aggarwal, and M. Salehi, “Deep learning for time series anomaly detection: A survey,” ACM Comput. Surv, vol. 57, no. 1, pp. 1–42, 2024.

[7] W. Wang, Z. Wang, Z. Zhou, H. Deng, W. Zhao, C. Wang, and Y. Guo, “Anomaly detection of industrial control systems based on transfer learning,” Tsinghua Sci. Technol, vol. 26, no. 6, pp. 821–832, 2021.

[8] J. Watts, F. Van Wyk, S. Rezaei, Y. Wang, N. Masoud, and A. Khojandi, “A dynamic deep reinforcement learning-Bayesian framework for anomaly detection,” IEEE Trans. Intell. Transp. Syst, vol. 23, no. 12, pp. 22884–22894, 2022.

[9] A. Guezzaz, Y. Asimi, M. Azrour, and A. Asimi, “Mathematical validation of proposed machine learning classifier for heterogeneous traffic and anomaly detection,” Big Data Min. Anal, vol. 4, no. 1, pp. 18–24, 2021.

[10] I. Siniosoglou, P. Radoglou-Grammatikis, G. Efstathopoulos, P. Fouliras, and P. Sarigiannidis, “A unified deep learning anomaly detection and classification approach for smart grid environments,” IEEE Trans. Netw. Serv. Manag, vol. 18, no. 2, pp. 1137–1151, 2021.

[11] H. Su, Z. Wu, H. Zhang, and Q. Du, “Hyperspectral anomaly detection: A survey,” IEEE Geosci. Remote Sens. Mag, vol. 10, no. 1, pp. 64–90, 2021.

[12] Q. Wang, K. Paynabar, and M. Pacella, “Online automatic anomaly detection for photovoltaic systems using thermography imaging and low rank matrix decomposition,” J. Qual. Technol, vol. 54, no. 5, pp. 503–516, 2022.

[13] M. Wang, Q. Wang, D. Hong, S. K. Roy, and J. Chanussot, "Learning tensor low-rank representation for hyperspectral anomaly detection," IEEE Trans. Cybern, vol. 53, no. 1, pp. 679–691, 2022.

[14] H. Gadde, "AI-driven anomaly detection in NoSQL databases for enhanced security," Int. J. Mach. Learn. Res. Cybersecur. Artif. Intell, vol. 14, no. 1, pp. 497–522, 2023.

[15] W. Hao, T. Yang, and Q. Yang, "Hybrid statistical-machine learning for real-time anomaly detection in industrial cyber–physical systems," IEEE Trans. Autom. Sci. Eng, vol. 20, no. 1, pp. 32–46, 2021.

[16] M. Gopalsamy, "Scalable anomaly detection frameworks for network traffic analysis in cybersecurity using machine learning approaches," Int. J. Curr. Eng. Technol, vol. 12, no. 6, pp. 549–556, 2022.

[17] M. Bahri, F. Salutari, A. Putina, and M. Sozio, "AutoML: State of the art with a focus on anomaly detection, challenges, and research directions," Int. J. Data Sci. Anal, vol. 14, no. 2, pp. 113–126, 2022.

[18] Y. Zhang, Y. Chen, J. Wang, and Z. Pan, "Unsupervised deep anomaly detection for multi-sensor time-series signals," IEEE Trans. Knowl. Data Eng, vol. 35, no. 2, pp. 2118–2132, 2021.

[19] M. Goswami, "AI-based anomaly detection for real-time cybersecurity," Int. J. Res. Rev. Tech, vol. 3, no. 1, pp. 45–53, 2024.

[20] K. Rezaee, S. M. Rezakhani, M. R. Khosravi, and M. K. Moghimi, "A survey on deep learning-based real-time crowd anomaly detection for secure distributed video surveillance," Pers. Ubiquitous Comput, vol. 28, no. 1, pp. 135–151, 2024.