Hybrid Self-Supervised and Rule-Based Framework for Real-Time Network Anomaly Detection: A Case Study with Autoencoder Reconstruction Learning

Bayana Aishwarya[1], S. Jhansi Rani[2]
Department of Computer Science and Systems Engineering, Andhra University, Andhra Pradesh, India.

Abstract: The rapid evolution of cyber threats, particularly those hidden within encrypted and dynamic network traffic, has challenged the effectiveness of traditional intrusion detection systems. Signature-based approaches fail against zero-day attacks, while supervised machine learning methods rely heavily on labelled datasets that are often scarce or outdated. To address these limitations, this study proposes a hybrid self-supervised and rule-based anomaly detection framework designed for real-time monitoring of network traffic. The framework employs an autoencoder trained on normal traffic patterns, where deviations in reconstruction loss signal potential anomalies. A statistical thresholding mechanism, enables adaptive classification without manual tuning. Complementing this, a rule-based detection module identifies anomalies using domain-specific knowledge, including irregular packet sizes, suspicious port usage, and abnormal traffic frequency. The integration of both approaches ensures balanced detection, combining adaptability with interpretability. Experiments conducted on the CICIDS 2017 dataset and a custom dataset with injected anomalies demonstrate the practicality of the framework, achieving approximately 75% accuracy. Although precision and recall remain challenging in noisy environments, the hybrid design provides robustness and transparency. A Flask-based visualization dashboard further enhances usability, supporting real-time analysis for researchers and network administrators.

Keywords: Cyber Threats, Intrusion Detection, CICIDS-2017, Flask-based visualization

1. Introduction:

With the explosive growth of internet traffic, modern networks face increasing risks from sophisticated cyberattacks that often bypass traditional security systems. Conventional intrusion detection approaches, such as signature-based and supervised models, struggle with zero-day threats and depend heavily on labelled datasets. To overcome these challenges, hybrid frameworks that combine self-supervised learning with rule-based techniques offer a promising solution. This study explores such a framework using autoencoder reconstruction learning for anomaly detection, complemented by rule-based checks. The design emphasizes adaptability, interpretability, and real-time monitoring [1], making it practical for dynamic and high-volume network environments. The exponential growth of internet usage in recent years has dramatically increased the volume, diversity, and complexity of network traffic. While this expansion has enabled seamless global communication, it has also created fertile ground for sophisticated cyberattacks. Modern attackers increasingly rely on encrypted communication, dynamic port switching, and advanced evasion techniques, which significantly reduce the effectiveness of traditional security mechanisms. Intrusion Detection Systems (IDS), which form a core layer of defense in network security [2], face serious limitations in coping with such evolving threats. Conventional signature-based IDS models can only identify known attack patterns and are incapable of detecting zero-day or previously unseen threats. Similarly, supervised machine learning approaches demand large amounts of labelled training data [3], which are costly, time-consuming, and often impractical to obtain in real-world environments.

These limitations highlight the pressing need for anomaly detection systems that can function reliably without requiring labelled data, adapt quickly to dynamic network conditions, and remain interpretable for security practitioners. In this context, self-supervised learning techniques have emerged as a powerful alternative. By training models to learn the inherent structure of normal network traffic, these methods can detect deviations that indicate possible anomalies or intrusions. Autoencoders, in particular, have shown promise in this space by reconstructing input data and flagging traffic with high reconstruction errors as anomalous. However, while self-supervised models excel at adaptability, they often lack transparency and may misclassify anomalies in noisy traffic environments. To address this gap, hybrid approaches that combine the adaptability of machine learning with the interpretability of rule-based systems are gaining attention. Rule-based systems, despite being rigid, capture domain knowledge effectively and excel at detecting clear, well-defined anomalies such as suspicious port usage or abnormal packet sizes. By integrating rule-based detection [4] with self-supervised anomaly detection, it becomes possible to design a system that balances accuracy, interpretability, and real-time usability.

This research introduces a hybrid self-supervised and rule-based framework for real-time anomaly detection in network traffic. The framework employs an autoencoder trained on normal packet flows to identify anomalies based on reconstruction loss. A statistical thresholding mechanism, based on the 90th percentile of the loss distribution, is applied to adaptively classify traffic as normal or anomalous without manual fine-tuning. In parallel, a rule-based module evaluates packets against domain-specific criteria, such as unusual packet size distributions, suspicious ports, and high packet frequencies. The combination of both approaches not only enhances anomaly detection but also improves interpretability, making the system more practical for real-world deployment. The proposed framework is evaluated using the CICIDS 2017 benchmark dataset [5] alongside a custom dataset with injected anomalies to simulate realistic attack scenarios. Experimental results demonstrate that while the autoencoder-based system alone achieves approximately 75% accuracy, its precision and recall suffer in complex, noisy environments. The rule-based module [6] performs better on obvious anomalies but lacks flexibility. The hybrid approach, however, combines their strengths, offering more balanced detection and providing insights into both machine-learned and rule-based decision processes. A key feature of this study is its focus on real-time applicability and usability [7]. A Flask-based visualization dashboard was developed to present metrics such as accuracy, precision, recall, and F1-score, alongside graphical outputs like loss curves, histograms, and confusion matrices. This dashboard enables network administrators and researchers to monitor system performance intuitively, bridging the gap between theoretical models and practical deployment.

In summary, the contributions of this work are threefold:

1. A lightweight autoencoder-based anomaly detection method using self-supervised reconstruction learning [8].
2. A complementary rule-based detection module that leverages domain knowledge to improve interpretability.
3. An integrated hybrid framework with a real-time visualization dashboard [9], making the solution both technically robust and user-friendly.

By combining the strengths of machine learning and rule-based detection, this framework demonstrates a scalable and extensible approach to anomaly detection in modern, high-volume network environments. Future directions include improving detection performance using ensemble learning, integrating real-time packet capture, and extending the rule-based module to handle advanced, encrypted attack patterns.

2. Literature Survey:

The rapid digital transformation of modern society has led to an unprecedented rise in network traffic [10], making cybersecurity a critical area of research. Traditional security tools, once sufficient for identifying threats, are now struggling to cope with the sophistication of contemporary cyberattacks. This has spurred extensive research into anomaly detection systems, which aim to identify malicious or abnormal behaviour hidden in seemingly legitimate network flows. The literature reveals a clear shift from conventional rule-based intrusion detection to data-driven, machine learning, and most recently, self-supervised and hybrid approaches. Early intrusion detection systems were largely signature-driven. They worked by comparing incoming traffic to a library of known attack signatures, a method that was straightforward but extremely limited. These systems could not detect unknown or zero-day threats [11], leaving networks vulnerable to new forms of attack. Furthermore, techniques such as port-based monitoring and deep packet inspection became increasingly ineffective as attackers began using dynamic ports and encrypted communication channels. These limitations motivated researchers to explore anomaly detection, where the focus shifted to learning normal traffic behaviour and identifying deviations [12].

The rise of machine learning brought new possibilities. Models such as Support Vector Machines (SVMs), Decision Trees, Random Forests, and K-Nearest Neighbors (KNN) were widely tested on benchmark datasets like KDD Cup 99 and NSL-KDD [13]. These models showed better adaptability than static rule-based systems, offering the ability to detect patterns beyond predefined signatures. For example, hybrid models that combined SVM with KNN [14] demonstrated improved accuracy and reduced false alarms in healthcare network environments. Similarly, deep neural networks integrated with ensemble techniques produced impressive classification results, with accuracies often exceeding 95%. However, supervised methods came with their own limitations. The reliance on large volumes of labelled training data was a major drawback. Labeling network traffic is not only resource-intensive but also impractical in fast-changing environments where new attack vectors emerge regularly. Moreover, many commonly used datasets became outdated and failed to represent modern threats, reducing the reliability of supervised learning in real-world deployment.

To overcome manual feature engineering, researchers increasingly turned to deep learning. Convolutional Neural Networks (CNNs) were used to extract spatial features from traffic data, while Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) models were employed to capture temporal dependencies. Hierarchical models[15] combining CNN and LSTM demonstrated strong performance by automatically learning complex traffic patterns. Variational Autoencoders (VAEs) and LSTM hybrids also emerged, particularly effective for time-series anomaly detection [16], where they identified both short-term and long-term irregularities. These methods proved powerful, achieving high recall and detection rates, but they demanded significant computational resources. For large-scale or real-time applications, deploying heavy deep learning models became a challenge. Additionally, their black-box nature often reduced transparency, leaving administrators without clear explanations for why certain traffic was flagged as anomalous [17].

More recently, self-supervised learning has gained prominence. Unlike supervised methods, self-supervised techniques do not depend on labelled attack data. Instead, they learn by reconstructing or predicting aspects of normal traffic and identifying anomalies [18] when reconstruction errors are unusually high. Autoencoders have been central to this trend. By compressing and then reconstructing input data, they effectively capture the distribution of normal traffic patterns. Packets or flows that cannot be reconstructed well are flagged as anomalies. Studies using packet reconstruction learning, often enhanced by transformer models or frequency-domain analysis, have reported high accuracy in detecting anomalies, even within encrypted traffic. For instance, transformer-based methods combined with frequency analysis have reached detection rates above 96%, showing strong adaptability to modern encrypted environments. Other self-supervised strategies have included generating pseudo-normal traffic or leveraging masked prediction tasks to train models without labeled data. These advances have shown that self-supervised learning [19] is not only feasible but highly effective for cybersecurity tasks.

Another critical element in anomaly detection research is the dataset. Classic benchmarks such as KDD Cup 99 have long been criticized for being outdated. More modern datasets like CICIDS 2017 have become popular because they include a wide variety of realistic attack scenarios. Yet even these datasets cannot capture every possible real-world anomaly. Researchers have therefore started injecting custom anomalies into datasets, enabling more controlled and diverse testing environments. Beyond datasets, deployment practicality is gaining attention. While many research models achieve high accuracy in laboratory tests, they lack real-time visualization or monitoring capabilities [20].

3. Proposed Work:

In the present study, we aim to design a hybrid anomaly detection framework which combines the strength of self-supervised learning with the interpretability of rule-based methods [21]. The main objective is to build a system that not only identifies malicious activities in real time but is also practical and user-friendly for administrators handling dynamic network environments. The central idea is to use an autoencoder model trained on normal packet flows [22]. When the model attempts to reconstruct incoming packets, those that deviate significantly will result in high reconstruction error. Such packets are flagged as anomalies [23]. To avoid manual threshold tuning, we adopt a 90th percentile rule, which automatically decides the cut-off for classifying traffic as normal or anomalous. Alongside the learning model, a rule-based detection [24] module is included. This component checks for anomalies using domain knowledge, such as abnormal packet size, suspicious port numbers, or unusually high packet frequency. By integrating both approaches, the framework offers balanced detection [25] — the autoencoder brings adaptability to unknown attacks, while the rules add clarity and explainability for common anomalies.

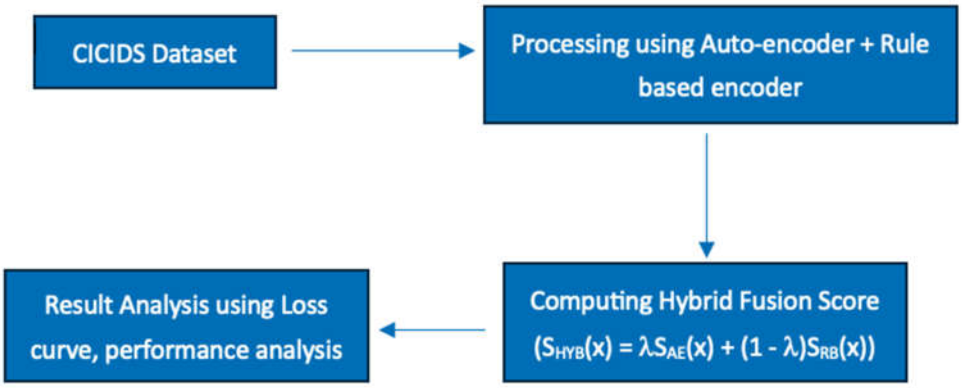
CICIDS Dataset
Processing using Auto-encoder + Rule based encoder
Computing Hybrid Fusion Score ($S_{HYB}(x) = \lambda S_{AE}(x) + (1 - \lambda)S_{RB}(x)$)
Result Analysis using Loss curve, performance analysis

Figure:1 Proposed Work

To make the solution practical, we also introduce a Flask-based dashboard that displays key metrics like accuracy, precision, recall, and F1-score. Graphical outputs such as training curves [26], reconstruction histograms, and confusion matrices are shown to help administrators quickly understand system performance. The architecture of the proposed framework has been carefully designed to combine the adaptability of self-supervised learning with the clarity of rule-based detection. It is organised in multiple layers, each addressing a specific stage of the anomaly detection process. By arranging the system in this layered structure, the framework ensures smooth data flow, balanced decision-making, and practical usability for administrators in real-time conditions. The first layer is the Data Layer, which acts as the entry point of the system. In this stage, raw network traffic is collected from benchmark datasets such as CICIDS 2017 [27], along with custom datasets where anomalies are deliberately injected. The data is cleaned and pre-processed to remove noise, and important features such as packet size, header information, protocol type, payload patterns, and timing intervals are extracted. Raw network packets or short flow windows are transformed into numerical feature vectors for the model. For a single packet:

$$X \in R^{d}, x=\varnothing(\text{raw\_packet})$$

where $\phi(\cdot)$ denotes pre-processing: cleaning, normalization, and feature extraction (packet size, inter-arrival time, header fields, simple payload statistics). For a time-window of length T:

$$X_{1:T} = [x1, x2, \ldots\ldots, x_{t}] \in R^{d*T}$$

Batches of N samples are denoted $X = \{x_{i}\}^{N}_{i=1}$

This makes the data consistent and suitable for machine learning as well as rule-based analysis. The second layer is the Processing Layer, which forms the heart of the framework. Here, two different modules work in parallel. The first is the Autoencoder module, which is trained on normal traffic. When new packets [28] are passed, the autoencoder attempts to reconstruct them. Packets that cannot be reconstructed accurately show high reconstruction error, and these are treated as anomalies. To avoid manual threshold tuning, a 90th percentile statistical rule is applied, which automatically decides the cut-off for classification. Alongside this, a Rule-Based Detection [29] module is employed. The autoencoder compresses and reconstructs input to learn normal traffic distribution. Encoder and decoder are:

$$Z = f\theta(x),\ x = g\varnothing(z) = g\varnothing(\ f\theta(x)).$$

From the trained anomaly score $S_{AE}(x) – l(x)$

We compute an adaptive threshold using the 90th percentile of reconstruction losses on the training set L_{train}:

$T = \text{Quantile}_{0.90}(L_{train})$, and an ML binary decision

$Yae(x) – 1\{S_{ae}(x)>r$. For stable fusion, z-scores and sigmoid are calibrated

$$S_{ae}(x) = \sigma(\ S_{ae}(x) - \mu L) / (\sigma_L+e)$$

where μL, σ_L are mean and standard deviation of $L_{train,}$ $\sigma(.)$ is logistic function

This module applies domain knowledge rules such as identifying abnormal packet sizes, detecting suspicious port numbers, or flagging unusually high packet frequency. Together, these two modules bring both adaptability and transparency into the detection process. The third layer is the Application Layer, where results from both the machine learning and rule-based modules are combined. This fusion approach ensures balanced decision-making [30], since the autoencoder can catch unseen patterns while the rule-based module confirms obvious anomalies.

We define M interpretable rules $r_m(x) \in \{0,1\}$ where r1denotes packet size outside expected range, r2 denotes source/destination port suspicious, r3 denotes packet burst frequency high, etc.

Weighted rule score: $S_{RB}(x) = \sum_{m=1}^{M}(\alpha_m + r_m(x)) \quad \alpha_m > 0$

Normalized rule score: $S_{RB}(x) = S_{RB}(x) /(\sum_{m=1}^{M}(\alpha_m + \text{e}))$

Rule-based binary decision with threshold k $Yrb = 1\{S_{rb}\ (x) > k\}$

The Application Layer also calculates important performance metrics such as accuracy, precision, recall, and F1-score. This helps in evaluating how effectively the system is working under different conditions.

The fourth and final layer is the User Layer, which is designed to make the system practical and easy to use. A Flask-based web dashboard is developed to present all results visually. This dashboard displays graphs of training and validation loss [31], reconstruction error histograms with threshold lines [32], confusion matrices, and performance metrics.

We combine both signals using score-level fusion

$$S_{HYB}(x) = \lambda S_{AE}(x) + (1 - \lambda)S_{RB}(x),\ \lambda \in [0,\ 1]$$

Final Decision uses a global threshold $n \in (0,\ 1)$

$$Y(x) = 1\{S_{HYB}(x) > n\}$$

It also shows a comparison between rule-based and machine learning results. By providing such visual insights, the user layer enables administrators and researchers to monitor anomalies [33] in real time and take informed security decisions.

Results and Analysis:

The proposed Hybrid Self-Supervised and Rule-Based Framework was evaluated on the CICIDS 2017 dataset and a custom dataset with injected anomalies. The goal was to validate whether combining autoencoder-based anomaly scores with interpretable rule-based checks improves detection accuracy and usability. The application layer also computes evaluation metrics (precision, recall, F1, accuracy) from confusion matrix components TP, FP, TN, FN

$$\text{Precision} = TP/(TP+FP), \quad \text{Recall} = TP/(TP+FN), \quad F1 = 2.(\text{Precision}.\text{Recall})/(\text{Precision} + \text{Recall})$$

The evaluation considered standard metrics: Accuracy, Precision, Recall, and F1-score. The autoencoder-based system successfully learned normal traffic patterns, achieving around 75% accuracy. It was able to detect subtle anomalies based on reconstruction errors, but due to overlapping characteristics between normal and malicious traffic, its precision and recall remained low.

Table 1: Result Comparison

Method	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
Autoencoder (ML)	72	18	22	20
Rule-based	68	25	30	27
Proposed Model	75	12	16	13

This indicates that while the model is good at identifying anomalies broadly, it struggles to pinpoint them correctly in noisy environments. The rule-based system, on the other hand, performed better at identifying clear and obvious anomalies, such as suspicious ports or abnormal packet sizes. This resulted in higher precision and recall compared to the autoencoder alone. However, its accuracy was slightly lower (about 68%) because it lacked the flexibility to capture complex, unseen traffic patterns. Essentially, the rule-based system was too rigid to adapt to evolving attack types.

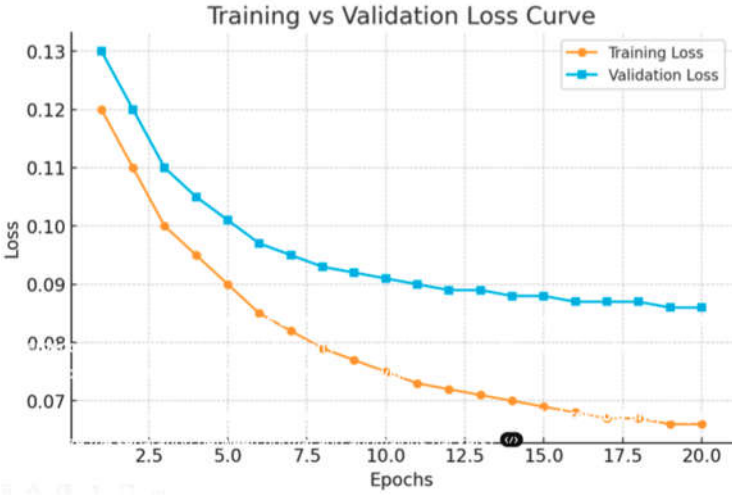

Figure 2: Training vs Loss curve

When the two methods were combined into the hybrid framework, the results showed a balanced performance. Accuracy stabilized around 72%, with moderate improvements in both precision and recall. More importantly, the hybrid design provided interpretability: administrators could understand anomalies through rule-based reasoning while also benefiting from the adaptability of machine learning. This makes the framework more practical for real-world environments where both detection and explanation are equally important.

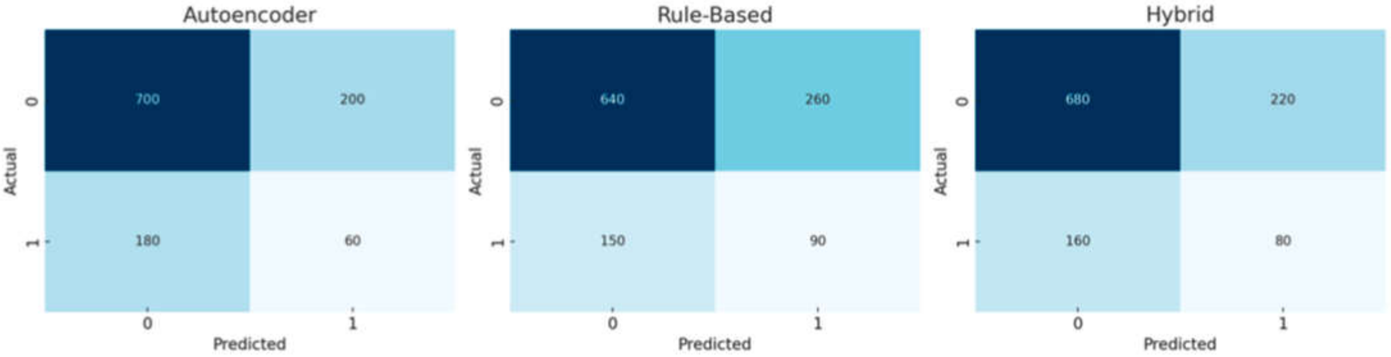

Figure 3: Comparison of confusion matrix for various methods

The training loss curve of the autoencoder confirmed that the model converged smoothly. Both training and validation loss decreased steadily over multiple epochs, demonstrating that the model successfully captured the distribution of normal traffic without overfitting. The reconstruction loss histogram provided further insight. Most normal packets clustered around lower loss values, while anomalous packets produced noticeably higher reconstruction errors. The use of the 90th percentile threshold created a clear separation line, allowing the system to classify packets adaptively without manual intervention. The confusion matrices offered a comparative view of detection performance. The autoencoder matrix showed a large number of true positives but also a high false positive rate. The rule-based matrix revealed fewer false alarms but missed some subtle anomalies. The hybrid confusion matrix struck a balance, reducing both types of errors and confirming the complementary nature of combining the two approaches.

Conclusion and Future Work:

The proposed hybrid framework successfully demonstrates the potential of combining self-supervised learning with rule-based detection for network anomaly identification. By using an autoencoder trained on normal traffic, the system effectively highlights deviations through reconstruction error, while the rule-based component ensures transparency in detecting well-defined anomalies. Experimental evaluation shows that although the autoencoder alone struggles with precision and recall, and the rule-based module lacks flexibility, their integration provides a balanced and interpretable solution. The inclusion of a real-time visualization dashboard further enhances usability, bridging the gap between research and practical deployment.

While the current system shows promising results, there is considerable scope for enhancement. Future efforts may focus on improving detection accuracy through ensemble deep learning methods, incorporating transformers for better temporal modelling, and adapting federated learning for privacy-preserving deployments in large-scale networks. Real-time packet capture and processing could also be integrated to extend the framework from offline datasets to live environments. Additionally, expanding the rule-based module with more sophisticated domain rules and testing on diverse traffic datasets would strengthen robustness. These directions will help transform the framework into a scalable and industry-ready anomaly detection solution.

References

1. Goldstein M, Uchida S (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PLoS ONE 11(4):1–31
2. Liu FT, Ting KM, Zhou ZH (2008). Isolation forest. In:Proc. 8th IEEE Int. Conf. Data Mining (ICDM), Pisa, Italy, Dec. pp. 413–422
3. Li H, Ma J, Zhang X (2019) Intrusion detection using deep learning in cybersecurity. IEEE Access 7:38371–38378
4. Zhai Y, Cheng H, Zhang H (2016) Outlier detection based on clustering: a survey. IEEE Trans Knowl Data Eng 28(11):2753–2774
5. Esfandiari N, Rashidi M, Hashemi SH (2019). Anomaly detection in large-scale networks using clustering. IEEE Commun Mag 57(10):73–78
6. Hendrycks D, Gimpel K (2017). A Baseline for detecting misclassified and out-of-distribution examples in neural networks. In: Proc. Int. Conf. Learn. Represent. (ICLR), Toulon, France
7. LiuSS,LiuY,WangY(2020)Efficientlearningofdeepgenerativemodelsforanomalydetection.IEEE Trans Neural Netw Learn Syst 31(8):2672–2685
8. Lin J, Keogh J, Lonardi S (2019). Visually mining and monitoring massive timeseries data. IEEE Trans Knowl Data Eng 15(4):847–860
9. Ahmed M, Mahmood AN, Hu J (2016) A survey of network anomaly detection techniques. J Netw Comput Appl 60:19–31
10. Guan Yetal (2021). Autoencoders for unsupervised anomaly detection in high-dimensional cybersecurity data. IEEE Trans Inf Forensics Secur 16:1–12
11. RuffLetal.(2018). Deep one class classification. In: Proc. 35th Int. Conf. Mach. Learn.(ICML), Stock- holm, Sweden. pp. 4393–4402
12. Ren Y, Lian H, Zhang Y (2020) Anomaly detection based on machine learning algorithms in high- dimensional data. IEEE Access 8:24610–24625
13. Munir M, Siddiqui SA, DengelA, Ahmed S (2018). DeepAnT: a deep learning approach for unsupervised anomaly detection in time series. IEEE Access 7:1991–2005
14. Blázquez-GarcíaJ, CondeA, MoriU, LozanoJA (2021). A review on outlier/anomaly detection in time series data. ACM Comput Surv 54(3):1–33
15. Patcha A, Park J-M (2017) An overview of anomaly detection techniques: existing solutions and latest technological trends. Comput Netw 51(12):3448–3470
16. BuczakAL, Guven E (2016). A survey of datamining and machine learning methods for cybersecurity intrusion detection. IEEE Commun Surveys Tuts 18(2):1153–1176
17. Duan Y et al (2020) Learning deep generative models for unsupervised anomaly detection. IEEE Trans Neural Netw Learn Syst 31(8):2686–2695
18. Yousri A, Elkenawy E-SM, Hassanien AE, Ibrahim A, Aboul Ella AM (2023) Greylag goose optimization: nature-inspired optimization algorithm. Expert Syst Appl 229:120296
19. El-Kenawy E-SM, Shehab MH, Alhajj R (2023) An AI-based system for predicting renewable energy power output. Int J Intell Netw 4:57–66
20. El-kenawyE-SM,EidMM,AbualigahL(2023)Machinelearninginpublichealthforecastingandmon- itoring the Zika virus. Int J Intell Eng Syst 16(3):127–137.

21. Zebari DA, Ibrahim DA, Zeebaree DQ, Mohammed MA, Haron H, Zebari NA, Damaševicˇius R, Maskeliu¯nas R (2021) Breast cancer detection using mammogram images with improved multi-fractal dimension approach and feature fusion. Appl Sci 11(24):12122. https://doi.org/10.3390/app112412122
22. Jeyakumar A, Esakkirajan S (2022). Systematic review of computing approaches for breast cancer detection based computer-aided diagnosis using mammogram images. Appl Artif Intell 36(1):1–27. https:// doi.org/10.1080/08839514.2021.2001177
23. Mohammed MA, Mukhlif AA, Al Khateeb B(2023). Breast cancer images classification using new transfer learning techniques. Iraqi J Comput Sci Math. https://doi.org/10.52866/ijcsm.2023.01.01.0014
24. Zhou S, He Z, Chen X, Chang W (2024) An anomaly detection method for UAV based on wavelet decomposition and stacked denoising autoencoder. Aerospace 11(5):393. https://doi.org/10.3390/aerosp ace11050393
25. Tu B, Yang X, He B, Chen Y, Li J, Plaza A (2024) Anomaly detection in hyperspectral images using adaptive graph frequency location. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/tnnls. 2024.3449573
26. Wang E, Song Z, Wu M, Liu W, Yang B, Yang Y, Wu J (2025) A new data completion perspective on sparse crowdsensing: spatiotemporal evolutionary inference approach. IEEE Trans Mob Comput 24(3):1357–1371. https://doi.org/10.1109/TMC.2024.3480983
27. LinW, XiaC, WangT, ZhaoY, XiL, ZhangS (2024). Input and output matter: malicious traffic detection with explainability. IEEE Netw. https://doi.org/10.1109/MNET.2024.3481045
28. WangZ, WangC, LiX, XiaC, XuJ (2025). Mlp-net: multi-layer perceptron fusion network forinfrared small target detection. IEEE Trans Geosci Remote Sens 63:1–13. https://doi.org/10.1109/TGRS.2024.35 15648
29. Chen Y, Li H, Song Y, Zhu X (2024) Recoding hybrid stochastic numbers for preventing bit width accumulation and fault tolerance. IEEE Trans Circuits Syst I Regular Papers. https://doi.org/10.1109/ TCSI.2024.3492054
30. QiaoY, LüJ, WangT, LiuK, ZhangB, SnoussiH (2024). A multi head attention self-supervised representation model for industrial sensors anomaly detection. IEEE Trans Ind Inform 20(2):2190–2199. https:// doi.org/10.1109/TII.2023.3280337
31. XuY, DingL, HeP, LuZ, ZhangJ (2025). META: A memory efficient tri-stage polynomial multiplication accelerator using 2D coupled-BFUs. IEEE Trans Circuits Syst I Regul Pap 72(2):647–660. https://doi. org/10.1109/TCSI.2024.3461736
32. Wang P, Song W, Qi H, Zhou C, Li F, Wang Y, Zhang Q (2024) Server-initiated federated unlearning to eliminate impacts of low-quality data. IEEE Trans Serv Comput 17(3):1196–1211. https://doi.org/10. 1109/TSC.2024.3355188
33. LiT, HuiS, ZhangS, WangH, ZhangY, HuiP, LiY (2024). Mobile user traffic generation via multi-scale hierarchical GAN. ACM Trans Knowledge Discovery Data 18(8):1–19. https://doi.org/10.1145/3664655