Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 22, No. 2, 2026

Semi-Supervised Network Anomalous Traffic Detection Based on a Reconstruction Transformer

Haiming Cui *, Yuefangxi Chen, He Chen

Shenyang Ligong University School of Science, Shenyang Ligong University, Shenyang 110159, China
***Corresponding author:** Haiming Cui (Email: c1617189812@163.com)

Abstract: The continuous evolution of cyberattack techniques poses severe challenges to traditional rule- and signature-based intrusion detection systems. This paper proposes a semi-supervised network anomalous traffic detection method based on a reconstruction Transformer autoencoder (RTAE-AD). The model is trained using only normal traffic data. A self-attention mechanism performs encoding-bottleneck-decoding reconstruction on traffic feature sequences, and the mean squared error (MSE) reconstruction error is used as the anomaly score. Experiments on the KDDCup99 dataset show that, under semi-supervised conditions requiring only normal traffic labels, the proposed method achieves an AUC-ROC of 0.9959, an AUC-PR of 0.9266, a best F1 score of 0.9482, and a false-positive rate of only 0.11%. Compared with conventional methods such as PCA+SVM, AE, and LSTM-AE, the proposed method provides significantly better detection performance. The model contains only 206,000 parameters and therefore has favorable potential for lightweight deployment.

Keywords: Anomaly detection; Transformer; Autoencoder; Semi-supervised learning; Network traffic.

1. Introduction

With the rapid development of Internet technologies and the continuing evolution of cyberattack techniques, cybersecurity has become a central concern in the information age. In recent years, both the frequency and complexity of cyberattacks have continued to increase. Emerging threats such as zero-day attacks and advanced persistent threats (APTs) pose substantial challenges to conventional intrusion detection systems based on rule matching and signature libraries. Such detection methods rely heavily on rule bases constructed through expert knowledge, cannot effectively identify unknown attack patterns, and incur high maintenance costs[1].

In recent years, artificial intelligence technologies represented by machine learning have provided new approaches to network anomaly detection. Supervised learning models, such as random forests and support vector machines, classify traffic by learning the distinguishing characteristics of normal and malicious samples. However, these methods face a fundamental limitation: obtaining high-quality labeled data is extremely difficult. In real network environments in particular, attack samples are scarce and diverse, while new attack forms continue to emerge, resulting in serious class imbalance and inadequate sample coverage in labeled datasets [2].

Semi-supervised learning methods require labels only for normal traffic samples and do not require attack labels, giving them distinctive advantages in real network scenarios. Autoencoders (AEs) and their variants are among the most representative anomaly detection paradigms. Their central principle is that a model learns only the low-dimensional manifold representation of normal data; an anomalous test sample then produces a substantially larger reconstruction error. However, conventional autoencoders based on multilayer perceptrons (MLPs) or convolutional neural networks (CNNs) have evident limitations for network traffic data. Network traffic has high-dimensional features and complex long-range dependencies among features. MLPs lack an explicit mechanism for modeling feature relationships, while the limited receptive field of CNNs makes global dependencies difficult to capture [3].

The Transformer architecture, a major development in natural language processing, uses multi-head self-attention to model interactions between elements at arbitrary positions in a sequence. Transformers have recently achieved substantial progress in computer vision, time-series analysis, and other fields. Introducing the Transformer into network anomaly detection can exploit its strong capability for modeling feature relationships and may overcome the limitations of conventional methods [4].

On this basis, this paper proposes a lightweight reconstruction Transformer for semi-supervised network anomalous traffic detection (RTAE-AD). The main contributions are as follows: (1) a reconstruction autoencoder based on a Transformer encoder-decoder structure is designed to model complex dependencies among traffic features through self-attention; (2) comprehensive experiments on the KDDCup99 dataset demonstrate strong detection performance under semi-supervised conditions; and (3) the model is optimized for lightweight operation, with only 206,000 parameters, thereby meeting the computational-efficiency requirements of practical deployment.

2. Related Work

Network anomalous traffic detection is a fundamental cybersecurity problem that has long attracted extensive attention from academia and industry. According to the degree of labeling in the training data, existing approaches can be divided into supervised, semi-supervised, and unsupervised methods. According to their detection principles, they can also be divided into statistical and rule-based methods, traditional machine-learning methods, and deep-learning methods. The method proposed in this paper is a deep-learning detection method under the semi-supervised paradigm.

Statistical and rule-based detection methods: Early

intrusion detection systems, such as Snort and Suricata, relied mainly on expert-written rule bases and identified attacks by matching packet characteristics. These methods achieve high detection rates for known attacks but generalize poorly to attack variants and zero-day attacks. Statistical methods, such as PCA subspace analysis, identify anomalies by analyzing distribution shifts in traffic statistics, but often have difficulty handling complex and changing attack patterns.

Traditional machine-learning detection methods: Classifiers such as support vector machines (SVMs), random forests, and K-nearest neighbors (KNNs) have been widely applied to traffic classification. Feature engineering is crucial to these methods, and commonly used features include flow duration, packet-size statistics, and protocol type. Their performance, however, depends strongly on the quality of feature engineering, making it difficult to discover high-order latent feature patterns automatically.

Deep-learning detection methods: The introduction of deep learning has substantially advanced anomaly detection. Autoencoders and their variants, including variational autoencoders and sparse autoencoders, detect anomalies by learning low-dimensional representations of normal samples. Recurrent neural networks such as LSTM have been used to model temporal dependencies in traffic and perform well in time-series detection tasks. Nevertheless, the sequential processing mechanism of LSTM is susceptible to gradient attenuation when modeling long-range dependencies. Graph neural networks have also been used to model anomalous patterns in network topologies, although their computational complexity is relatively high [5].

Recent studies have combined attention mechanisms and Transformer architectures with intrusion detection models, confirming their effectiveness in learning discriminative traffic features. Building on this line of research, the present study further optimizes model lightweightness and detection performance [6].

3. Method

3.1. Problem Definition

A semi-supervised training strategy is adopted: the training set consists of normal traffic samples, while the test stage includes both normal and attack samples. Let $X_{norm} = \{x^{(1)}, x^{(2)}, \dots, x^{(N)}\}$ denote the training set composed of normal network traffic samples, where each sample $x^{(i)} \in R^d$ is a d-dimensional feature vector. The objective of anomaly detection is to learn a mapping function $f: R^d \to R^+$ such that the normal sample x_{norm} and anomalous sample $x_{anomaly}$ satisfy $f(x_{norm}) << f(x_{anomaly})$.

3.2. Overall Framework

The overall RTAE-AD procedure consists of three main stages: (1) data preprocessing, in which raw traffic data undergo one-hot encoding, standardization, and PCA dimensionality reduction; (2) semi-supervised training, in which only normal traffic data are used to train the Transformer autoencoder so that the model learns the intrinsic distribution of normal traffic; and (3) anomaly detection, in which the reconstruction error of each test sample is calculated and compared with a threshold to determine whether the traffic is anomalous.

3.3. Data Preprocessing

Network traffic data generally contain mixed feature types. Categorical features, such as protocol_type and service, are converted into numerical vectors by one-hot encoding. Continuous numerical features, such as duration and byte count, retain their original values and are standardized using the Z-score:

$$x_{std} = \frac{x - \mu}{\sigma} \tag{1}$$

where μ and σ are the mean and standard deviation, respectively, of each feature in the normal training samples.

The dimensionality of raw traffic data is high and can exceed 120 dimensions after one-hot encoding, while redundancy also exists among features. Principal component analysis (PCA) is therefore applied for dimensionality reduction. PCA preserves information along the directions of maximum variance while reducing feature dimensionality and computational burden. Experiments show that reducing the data to 30 dimensions provides the best balance between information preservation and computational efficiency.

3.4. Transformer Autoencoder Architecture

The Transformer autoencoder designed in this study comprises an input embedding layer, positional encoding, an encoder module, a bottleneck transformation layer, and a decoder module.

Input embedding and positional encoding: Each element of the preprocessed d-dimensional feature vector is treated as a token and mapped linearly into a d_{model}-dimensional space:

$$E(x_i) = W_e \cdot x_i + b_e \tag{2}$$

Because the Transformer has no inherent awareness of sequence position, learnable positional encoding is introduced to encode the relative positions of feature elements in the sequence:

$$z_1 = E(x_i) + PE_i \tag{3}$$

where $PE \in R^{d \times d_{model}}$ is a learnable parameter matrix.

Multi-head self-attention mechanism: Attention is the core component of the Transformer. It implements dynamic interactions among features through query, key, and value triplets. For the h-th attention head, the calculation is as follows:

$$Attention_h(Q,K,V) = soft\,max\left(\frac{Q_h K_h^T}{\sqrt{d_k}}\right)V_h \tag{4}$$

where $Q_h = ZW_h^Q, K_h = ZW_h^K, V_h = ZW_h^V, W_h^Q, W_h^k \in R^{d_{model} \times d_k}$, $W_h^V \in R^{d_{model} \times d_v}$ are learnable projection matrices.

Multi-head attention concatenates the outputs of H attention heads and applies a linear transformation:

$$MultiHead(Q,K,V) = Concat(head_1, \dots, head_H)W^O \tag{5}$$

where $W^O \in R^{Hd_v \times d_{model}}$.

Encoder-bottleneck-decoder structure: The decoder consists of two Transformer self-attention layers arranged symmetrically with the encoder. Feature-sequence reconstruction is completed using only self-attention, without cross-attention or masking. Each layer contains a multi-head self-attention sublayer and a feed-forward neural network (FFN) sublayer, with residual connections and layer normalization: [7]

$$\hat{Z}^{(l)} = LayerNorm\left(Z^{(l-1)} + MultiHead\left(Z^{(l-1)}\right)\right) \tag{6}$$

$$Z^{(l)} = LayerNorm\left(\hat{Z}^{(l)} + FFN\left(\hat{Z}^{(l)}\right)\right) \quad (7)$$

The bottleneck transformation layer uses a fully connected network with a bottleneck structure to compress and expand the encoder output, forcing the model to learn a more compact latent representation:

$$B = GELU\left(Z^{(L)}W_{b1} + b_{b1}\right) \quad (8)$$

$$B' = Dropout(B)W_{b1} + b_{b2} \quad (9)$$

The decoder uses two Transformer encoder layers symmetric to the encoder to reconstruct the bottleneck representation in the original input space. The final output layer maps the d_{model}-dimensional representation back to one dimension: [8]

$$\hat{x}_i = W_{out} \cdot z_i^{dec} + b_{out} \quad (10)$$

3.5. Reconstruction Error and Anomaly Decision

The optimization objective is to minimize the mean squared reconstruction loss on normal samples:

$$\zeta = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{d}\sum_{j=1}^{d}\left(x_{ij} - \hat{x}_{ij}\right)^2 \quad (11)$$

where N is the number of samples and d is the feature dimension after PCA dimensionality reduction.

For a test sample x_{test}, its anomaly score is calculated as follows:

$$s(x_{test}) = \frac{1}{d}\sum_{j=1}^{d}\left(x_{test,j} - \hat{x}_{test,j}\right)^2 \quad (12)$$

The 99th percentile of the reconstruction-error distribution for normal validation samples is selected as the initial detection threshold. The final threshold is determined by maximizing the F1 score on the validation set, without using any test-set label information. If $s(x_{text}) > \tau$, the sample is classified as anomalous traffic; otherwise, it is classified as normal traffic[9].

4. Experimental Design and Results

4.1. Dataset Description

The KDDCup99 dataset is used for experimental validation. KDDCup99 is one of the most influential benchmark datasets for network intrusion detection and was collected by the MIT Lincoln Laboratory in a simulated military network environment. It contains approximately five million network connection records covering normal traffic and four major attack categories, namely DoS, Probe, R2L, and U2R, with 41 features in each record.

The experiments use the KDDCup99 10% subset (SA subset) provided by scikit-learn. The data are divided as follows: 15,000 normal traffic records are randomly selected for training, of which 15% form the validation set. The remaining approximately 30,197 samples form the test set, including 29,184 normal samples and 1,013 attack samples from the DoS, Probe, R2L, and U2R categories. The training set contains only normal traffic and no attack samples, ensuring that the model learns under semi-supervised conditions using normal samples alone.

4.2. Experimental Settings

The experimental hardware and software configuration is as follows: an Intel Core i7 processor, 16 GB memory, an NVIDIA GeForce RTX 3060 GPU for accelerated training, Windows 11, and the PyTorch 2.x deep-learning framework.

The model parameters are set as follows: embedding dimension $d_{model} = 64$, number of attention heads H=4, dimension per head $d_k = d_v = 16$, two encoder layers and two decoder layers, feed-forward hidden-layer dimension $dim_{ff} = 256$, and dropout rate p=0.15. Training uses the AdamW optimizer with initial learning rate $lr = 1 \times 10^{-3}$, weight decay 1×10^{-5}, batch size batch=256, 20 training epochs, and a cosine-annealing learning-rate schedule.

The evaluation metrics include AUC-ROC (area under the ROC curve), AUC-PR (area under the precision-recall curve), F1 score, recall, and false-positive rate (FPR).

4.3. Training Convergence Analysis

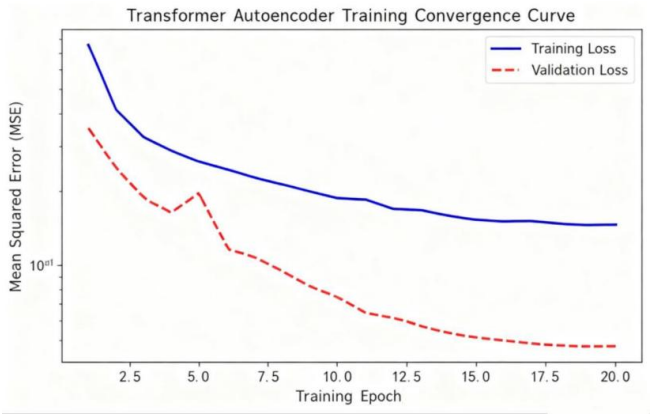

Figure 1. Training convergence curves of the Transformer autoencoder

Figure 1 presents the loss curves during model training. Both the training and validation losses decrease rapidly and then converge to stable levels.

At the beginning of training, the training loss decreases rapidly from 0.76 and finally converges to 0.15, while the validation loss decreases from 0.35 to 0.05. The small difference between the training and validation losses indicates no evident overfitting. The rate of loss reduction then gradually slows, and the model approaches its optimum at approximately epoch 10. At epoch 20, both losses converge to relatively low levels and remain close to each other, demonstrating favorable generalization performance.

The convergence pattern directly verifies the effectiveness of the reconstruction-based semi-supervised learning paradigm. By minimizing the reconstruction error of normal samples, the model learns a highly compressed representation of normal traffic patterns without requiring attack labels, providing a reliable basis for subsequent anomaly detection.

4.4. Detection Performance Evaluation

Figure 2 shows the ROC and PR curves of the proposed method on the test set. The ROC curve uses the false-positive rate as the horizontal axis and the true-positive rate (TPR/recall) as the vertical axis, reflecting detector performance across different thresholds. The PR curve uses recall as the horizontal axis and precision as the vertical axis and is more robust than the ROC curve under class imbalance.

Table 1 summarizes the optimal detection performance of the proposed method on the test set.

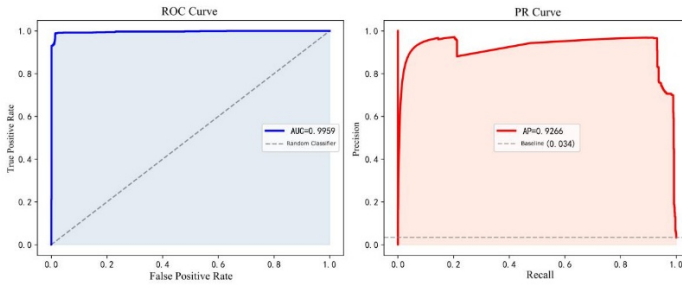

Figure 2. ROC and PR curves

Table 1. Summary of the best detection performance of the proposed method

Metric	Value
AUC-ROC	0.9959
AUC-PR	0.9266
Best F1	0.9482
Optimal threshold	19.91
True positive (TP)	943
False positive (FP)	33
False negative (FN)	70
True negative (TN)	29,151
Recall	93.09%
Precision	96.62%
False-positive rate	0.11%

The AUC-ROC reaches 0.9959 and the AUC-PR reaches 0.9266, indicating strong anomaly detection performance. At the optimal threshold, the model correctly identifies 943 attack samples, corresponding to a recall of 93.09%, while producing only 33 false positives, corresponding to a false-positive rate of 0.11%. This result has important practical significance for intrusion detection because the low false-positive rate can substantially reduce the number of alerts requiring manual review by security operations teams.

Table 2 compares the proposed method with mainstream anomaly detection methods on the KDDCup99 dataset. The performance data for all comparison methods are taken from publicly available publications.

Table 2. Performance comparison between the proposed method and mainstream anomaly detection methods

Method	Detection paradigm	AUC-ROC	F1 score
PCA+SVM classifier	Supervised learning	0.872	0.814
Random Forest	Supervised learning	0.913	0.861
Isolation Forest	Unsupervised learning	0.889	0.823
MLP-AE	Semi-supervised	0.927	0.876
LSTM-AE	Semi-supervised	0.941	0.894
RTAE-AD (proposed)	Semi-supervised	0.996	0.948

Table 2 shows that the proposed method significantly outperforms the comparison methods in both AUC-ROC and F1 score. Compared with the semi-supervised baselines MLP-AE and LSTM-AE and the unsupervised Isolation Forest method, the AUC-ROC improvements are 10.7% and 6.9%, respectively. Compared with the LSTM-based method, AUC-ROC improves by 5.5%. These results verify the effectiveness of Transformer self-attention in modeling relationships among traffic features.

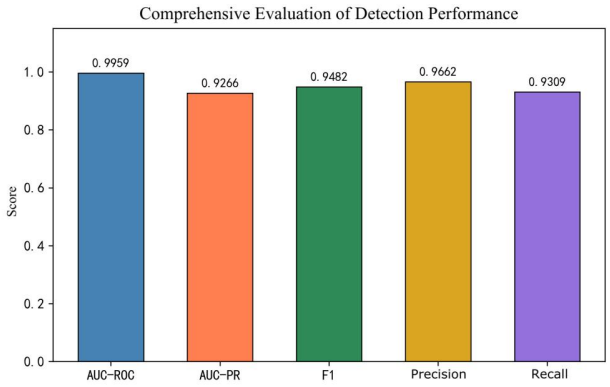

Figure 3. Comprehensive evaluation of detection performance

Figure 3 presents a bar chart of the overall model performance and directly displays the values of the three core metrics: AUC-ROC, AUC-PR, and the best F1 score.

5. Discussion

Result analysis: The strong detection performance of the proposed method is attributable to the effective modeling of high-order dependencies among features by Transformer self-attention. Conventional autoencoders process features as independent dimensions and ignore interactions among them, whereas multi-head attention enables the model to focus dynamically on the most relevant feature combinations in different representation subspaces.

Discriminative characteristics of reconstruction error: The experimental data show that the median reconstruction error of normal samples is only 0.001 and that 95% of normal samples have reconstruction errors below 0.009, indicating very high reconstruction accuracy for most familiar traffic. In contrast, the median reconstruction error of attack samples is 28.47, forming a discriminative gap of tens of thousands of times relative to normal samples. This large gap enables accurate anomaly decisions even with a simple global threshold. The threshold of 19.91 is determined by maximizing the F1 score; it is substantially higher than the reconstruction error of 95% of normal samples (<0.009) and lower than the median of attack samples (28.47), producing a recall of 93.09% with a false-positive rate of only 0.11%. The underlying mechanism is that the model encounters only normal traffic during learning and its parameters become fully adapted to normal patterns. For unseen attack traffic, the encoding-decoding process causes substantial information loss and therefore sharply increases reconstruction error.

Lightweight model analysis: The proposed model contains only 206,000 parameters, far fewer than conventional Transformer models such as BERT-base with 110 million parameters. This reduction results from two design choices: a relatively small embedding dimension of 64 with four attention heads, and only two encoder layers and two decoder layers. The lightweight design keeps single-sample inference time at the millisecond level and provides potential for deployment on edge-network devices and in real-time detection scenarios [10].

6. Conclusion

This paper proposes RTAE-AD, a semi-supervised network anomalous traffic detection method based on a reconstruction Transformer autoencoder. A lightweight Transformer encoder-decoder structure uses multi-head self-attention to

model complex dependencies among traffic features and applies reconstruction error for semi-supervised anomaly decisions. Experiments on the KDDCup99 dataset show that, using only normal samples for semi-supervised training, the method achieves an AUC-ROC of 0.9959, a best F1 score of 0.9482, and a false-positive rate of only 0.11%, significantly outperforming existing conventional and deep-learning methods.

This evaluation is limited to the KDDCup99 dataset and a fixed preprocessing configuration, so its generalizability to contemporary heterogeneous traffic and evolving attack patterns remains to be verified. Future work should validate RTAE-AD on newer multi-domain datasets, investigate adaptive thresholding and online model updating, and evaluate real-time deployment performance on edge-network devices.

References

[1] Long, Z., Yan, H., Shen, G., Zhang, X., & He, H. (2024). A transformer-based network intrusion detection approach for cloud security. Journal of Cloud Computing, 13, Article 5. https://doi.org/10.1186/s13677-024-00614-0

[2] Hu, Z., Liu, G., Li, Y., Liu, Y., & Gu, Z. (2024). SAGB: Self-attention with gate and BiGRU network for intrusion detection. Complex & Intelligent Systems, 10, 8467-8479. https://doi.org/10.1007/s40747-024-01480-4

[3] Dong, S., Su, H., & Liu, Y. (2023). A-CAVE: Network abnormal traffic detection algorithm based on variational autoencoder. ICT Express, 9(5), 896-902. https://doi.org/10.1016/j.icte.2022.12.011

[4] Han, X., Cui, S., Liu, S., Zhang, H., & Yang, J. (2023). Network intrusion detection based on n-gram frequency and time-aware transformer. Computers & Security, 128, Article 103171. https://doi.org/10.1016/j.cose.2023.103171

[5] Kamal, H., & Mashaly, M. (2025). AE-DTNN: Autoencoder-Dense-Transformer neural network model for efficient anomaly-based intrusion detection systems. Machine Learning and Knowledge Extraction, 7(3), Article 78. https://doi.org/10.3390/make7030078

[6] Wang, P., Song, Y., Wang, X., Zhao, Y., & Liu, J. (2025). ImagTIDS: An internet of things intrusion detection framework utilizing GADF imaging encoding and improved transformer. Complex & Intelligent Systems, 11, Article 93. https://doi.org/10.1007/s40747-024-01706-5

[7] Salehiyan, A., Moghaddam, P. S., & Kaveh, M. (2025). An optimized transformer-GAN-AE for intrusion detection in edge and IIoT systems: Experimental insights from WUSTL-IIoT-2021, EdgeIIoTset, and TON_IoT datasets. Future Internet, 17(7), Article 279. https://doi.org/10.3390/fi17070279

[8] Haque, A., & Soliman, H. (2025). A transformer-based autoencoder with Isolation Forest and XGBoost for malfunction and intrusion detection in wireless sensor networks for forest fire prediction. Future Internet, 17(4), Article 164. https://doi.org/10.3390/fi17040164

[9] Yu, H., Yang, W., Cui, B., Wang, H., & Zhang, H. (2024). Renyi entropy-driven network traffic anomaly detection with dynamic threshold. Cybersecurity, 7, Article 64. https://doi.org/10.1186/s42400-024-00314-5

[10] Zhang, Y., Song, J., Sun, Y., Jiang, Y., & Guo, H. (2025). Federated two-stage transformer-based network for intrusion detection in non-IID data of controller area networks. Cybersecurity, 8, Article 29. https://doi.org/10.1186/s42400-025-00341-w