CIT. Journal of Computing and Information Technology, Vol. 34, No. 2, June 2026, 81–98
doi: 10.20532/cit.2026.1006347

Optimization of Incremental Network Intrusion Real-Time Detection Algorithm Based on Computer Data Simulation

Hui Xiong, Chao Yuan and Ping Zhao

Sichuan Vocational and Technical College, Suining City, Sichuan Province, China

High-quality and representative network intrusion data is the basis for building a network intrusion detection system. Given the current situation in which the use of traditional algorithms is difficult to satisfy the demands of real-time detection, this paper constructs a network intrusion data simulation method based on LDMs, and combines it with the improved LST model to perform real-time detection of incremental network intrusions. The experiments demonstrated that the LST model outperformed the comparison models in key indicators such as detection rate, precision rate, recall rate, and F1-score, and showed significant advantages in running time. After the specific LST model converged, its accuracy, recall, and F1-score were 0.983, 0.971, and 0.958, respectively. When dealing with advanced persistent threat attacks and zero-day attacks, the detection rate reached 97.4% and 96.8%, and the running time was 160.3 s and 155.2 s, respectively. The LST model not only has high accuracy in detecting complex network intrusion behaviors, but also has obvious advantages in operating efficiency. This paper provides a strong guarantee for network security, helps to promptly discover and prevent potential network threats, and protects the stable operation of network systems.

ACM CCS (2012) Classification: Security and privacy → Network security → Denial-of-service attacks

Keywords: computer data simulation, incremental update, network intrusion, real-time monitoring, LST model

1. Overview

In today's digital era, the rapid development of network technology not only promotes global interconnection, but also brings many network security challenges. Network intrusions have already posed a serious threat to the integrity, confidentiality and availability of information systems [1-2]. As the network environment becomes increasingly complex and attack methods continue to evolve, traditional network security protection measures are no longer able to effectively deal with these threats. Incremental network refers to a network environment that continues to dynamically evolve in structure and data flow. Its core characteristics are that network traffic, node connections, behavioral patterns, and threat vectors are continuously generated, updated, and accumulated over time. Its continuous and fluid data growth model makes it difficult for traditional detection methods based on full historical data or fixed rules to achieve low-latency and adaptive identification of emerging attacks. As a key line of defense for network security, intrusion detection systems can monitor and identify abnormal behaviors in the network in real time, which is crucial to ensuring network security.

At this stage, the development of Network Intrusion Detection (NID) systems faces many challenges. Since the increasing complexity and concealment of network intrusion behaviors, conventional detection methods based on rules and signatures are difficult to deal with new attacks [3,4]. Secondly, the rapid growth of network traffic and the high-dimensional characteristics of data make the system face huge computing pressure when processing large-scale data [5,6]. In addition, there is

a difficult-to-balance contradiction between the real-time performance and accuracy of the intrusion detection system. High real-time requirements often come at the expense of detection accuracy, and high-precision detection may lead to response delays. In addition, the system has insufficient adaptability, making it difficult to quickly respond to changes in the network environment and the emergence of new attacks [7,8].

This study designs a Real-Time Detection of Network Intrusion (RTDNI) method based on Latent Diffusion Models (LDMs) and improved Location-Sensing Transformer (LST). This study aims to build a high-quality Network Intrusion Data (NIdata) simulation system and optimize the detection algorithm in the NID system to enhance the intrusion system's adaptability to new attacks. The innovation points of this research are as follows: Firstly, by introducing conditional information and adaptive noise scheduling in LDMs, it achieves high-fidelity and high-efficiency data generation for incremental threats, solving the problem of scarce and lagging real intrusion data. Secondly, by deeply integrating atrous convolution and incremental update mechanism into the LST, a real-time detection model capable of continuously and efficiently adapting to changes in data streams is designed. The data generation and detection models are not isolated improvements but are jointly designed through the incremental paradigm, thereby enhancing the system's real-time perception and adaptation capabilities for new and complex attacks in dynamic networks.

2. Literature Review

Scholars have conducted significant research on NID systems. AboulEla and Kashef proposed a comprehensive framework model that integrates large language models, graph neural networks, and explainable artificial intelligence in response to the current situation where NID systems face increasingly complex cyber threats. This framework could significantly improve the accuracy, interpretability, and real-time detection capabilities of NID [9]. Alshehri *et al.* proposed a deep convolutional neural NID model built on the Self-Attention Mechanism (SAM) in view of the increasingly complex security threats faced by industrial Internet of Things networks. The model's detection performance on datasets such as IoT-ID20 and Edge-IoTset was better than traditional machine learning (ML) and deep learning (DL) methods. It could effectively address poor model generalization ability caused by data imbalance and repeated data [10]. Imrana *et al.* put forth a NID system built on two-layer feature fusion to address the current situation of insufficient detection accuracy of traditional NID systems when processing high-dimensional data and complex attack patterns. This model combined CNN and GRU for spatial and temporal feature extraction. The model achieved detection accuracy of 99.86% and 99.54% on the NSL-KDD and UNSW-NB15 datasets, significantly better than traditional ML methods and existing DL models [11]. He *et al.* proposed a hybrid model based on conditional variational autoencoder GAN and 1D CNN in response to the current situation of category imbalance in NID systems. The model achieved detection accuracy of 90.11% and 88.86% on the NSL-KDD and UNSW-NB15 datasets, significantly better than the traditional class balancing method and existing intrusion detection systems [12].

Although the above studies have achieved good experimental results, the model's generalization capacity has not been effectively validated, and the computational efficiency needs to be further improved. Considering that a NID system has real-time challenges and high resource consumption, this study proposes an RTDNI method that combines LDMs and LST models. It aims to enhance the application value of NID systems in complex network environments, thereby providing strong support for network security protection.

3. Methodology

3.1. NIdata Simulation Construction Based on LDMs

In the context of increasingly complex network environments and evolving network intrusion threats, obtaining high-quality and representative NIdata is crucial to building an effective intrusion detection system. Traditional data collection methods are often limited by issues such as

data timeliness, completeness, and difficulty in acquisition, making it difficult to meet the needs of real-time detection algorithm optimization [13,14]. Considering that LDMs can learn complex data distributions through multiple steps of diffusion and inverse processes, they can generate network traffic data with high perceptual quality, rich features, and diverse patterns. This is crucial for simulating the ever-changing attack behaviors. Therefore, this study proposes a method based on LDMs to construct an NIdata simulation system to provide rich and accurate data support for the optimization of the incremental RTDNI algorithm. LDMs are a type of generative model based on the diffusion process. The core idea is to convert the data from the target distribution to a Gaussian distribution by adding noise, and then restore the Gaussian distribution to the target distribution through the process of learning inverse diffusion. The structure based on LDMs is shown in Figure 1.

In Figure 1, the forward diffusion process is mainly a Markov chain, which gradually adds noise to data x_0 and finally converts it into Gaussian noise. The reverse generation process uses a neural network to predict the noise at each step, thereby gradually recovering the data. The mathematical expression of the forward diffusion process is shown in formula (1).

$$x_{t+1} = \sqrt{1-\beta_t}\,x_t + \sqrt{\beta_t}\,\varepsilon_t \tag{1}$$

In formula (1), β_t is the predefined noise scheduling parameter. ε_t is standard Gaussian noise. The goal of the reverse generation process is to generate a neural network model ε_θ to satisfy formula (2).

$$x_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\left(x_t - \sqrt{\beta_t}\,\varepsilon_\theta\left(x_t, t\right)\right) \tag{2}$$

For LDMs, the relevant expression of training the model by minimizing the loss function L is shown in formula (3).

$$L = E_{t,x_0,\varepsilon_t}\left[\left\|\varepsilon_t - \varepsilon_\theta\left(x_t, t\right)\right\|^2\right] \tag{3}$$

In formula (3), E is the expected value. Although LDMs perform well in data generation, there is still the problem of poor data correlation in NIdata simulations. To better adapt to the needs of NID, this study introduces condition information and noise scheduling to optimize LDMs. After introducing conditional information such as network protocol type and attack type, the relevance and pertinence of the data can be significantly improved. This study uses the conditional diffusion model to achieve this process [15,16]. The reverse generation process based on the conditional diffusion model is shown in formula (4):

$$x_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\left(x_t - \sqrt{\beta_t}\,\varepsilon_\theta\left(x_t, t, c\right)\right) \tag{4}$$

Here, c is conditional information. By introducing conditional information, the model can learn to generate data that is more consistent with specific scenarios given the conditional information.

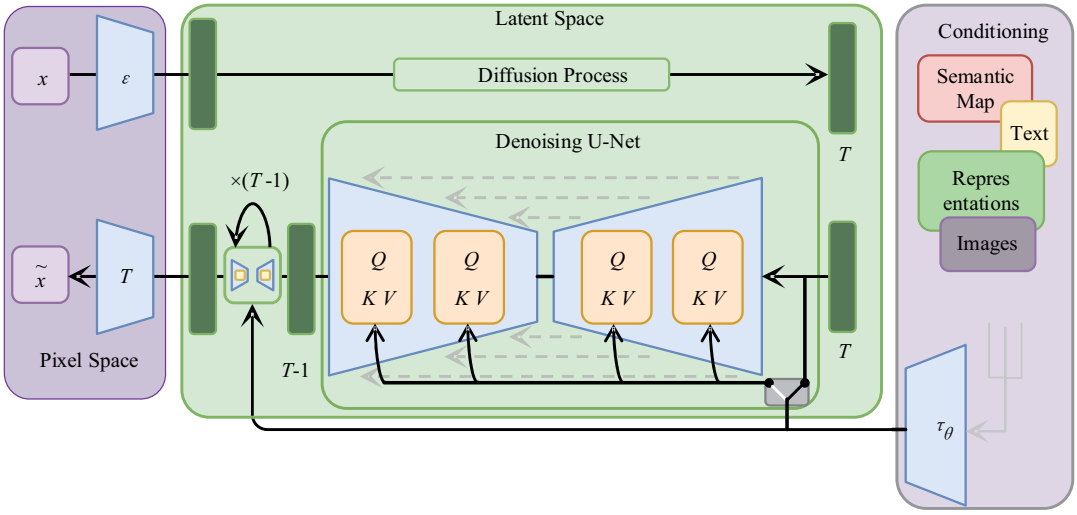


Figure 1. Structural diagram of an LDM.

Traditional LDM models use a fixed noise schedule β_t, which may lead to lower generation efficiency in some cases. To improve the generation efficiency, this study further introduces the ANS method, which dynamically adjusts the noise parameters according to the complexity of the data [17,18]. ANS can be implemented based on formula (5):

$$\beta_t = \beta_{t-1} + \alpha \cdot R(x_{t-1}) \tag{5}$$

Here, α is an adjustment parameter, and $R(x_{t-1})$ is the variance at step $t-1$. By introducing ANS, the model can dynamically adjust noise parameters according to the complexity of the data, thereby ensuring generation quality while improving generation efficiency. Figure 2 shows the flowchart of NIdata simulation construction.

In Figure 2, the target distribution characteristics of NIdata are first clarified, including the typical characteristics of normal network traffic and various intrusion behaviors. Appropriate initial data are selected, the noise scheduling parameters are initialized, and noise is added to make it closer to the Gaussian distribution. The noise addition process at each step is controlled by a Markov chain to ensure that the data gradually loses its original features. A neural network ε_θ is constructed to learn the reverse diffusion process. Introducing conditional information enables the model to generate data that meets specific scenarios. The ANS method is adopted to dynamically adjust the noise parameters according to the complexity of the data. Optimizing the model parameters by minimizing the difference between the prediction error and the true noise allows the model to accurately recover from the Gaussian distribution to the target distribution [19,20]. Finally, after the model training is completed, the trained inverse generation model is used to gradually restore data from the Gaussian distribution to generate NIdata that conforms to the target distribution.

3.2. Incremental RTDNI Method Based on LST Model

After successfully constructing the NIdata simulation system based on LDMs, based on this data, this study proposes an incremental RTDNI method based on the improved LST. The LST is a neural network model built on the Transformer architecture, which can be used to process data with spatial location information [21,22]. The basic structure of LST mainly includes key parts such as position coding, multi-head SAM, feed-forward neural network, residual connection, and layer normalization. The structure based on the LST model is shown in Figure 3.

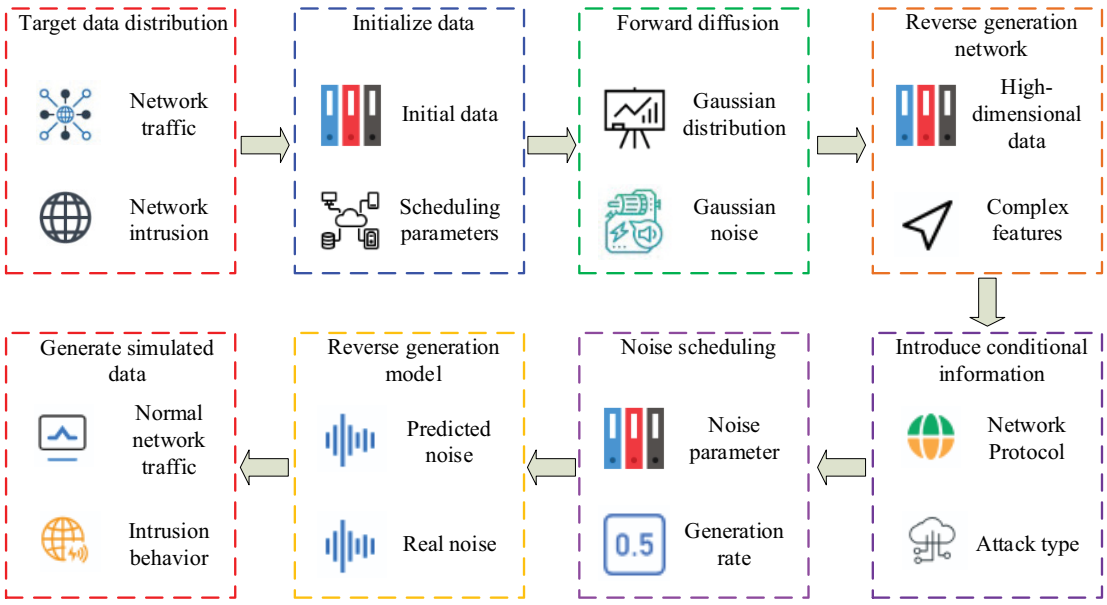


Figure 2. The construction process of an NIdata simulation.

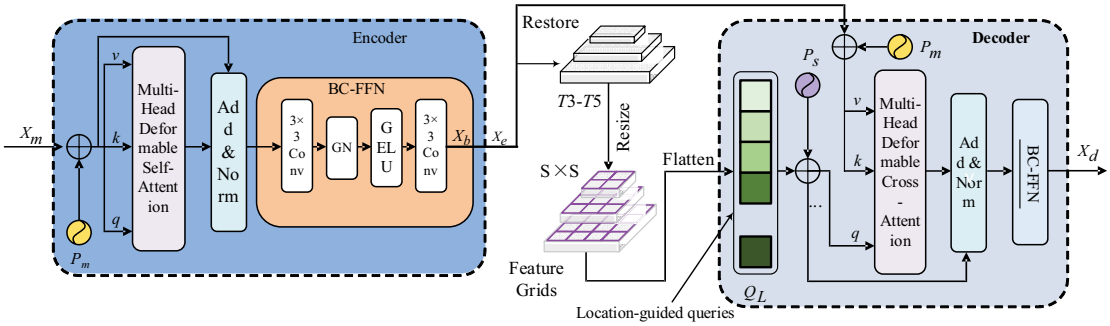


Figure 3. Structure of the proposed LST model.

In Figure 3, position encoding is used to provide the model with spatial location information for each data point. The relevant calculation of position encoding is shown in formula (6):

$$\begin{cases} PE(pos,2i)=\sin\left(\dfrac{pos}{10000^{2i}/d_m}\right) \\ PE(pos,2i+1)=\cos\left(\dfrac{pos}{10000^{2i}/d_m}\right) \end{cases} \quad (6)$$

Here, pos is the position, i is the dimension, and d_m is the dimension of the model. The multi-head SAM allows the model to learn information in parallel in subspaces of different representations. The self-attention calculation of each head is given by formula (7):

$$Attention(Q,K,V)=\text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (7)$$

Q, K, and V are query, key, and value matrices. d_k denotes the dimension of the key. In the LST model, each Transformer layer contains a feedforward neural network (FN), as shown in formula (8):

$$FN(x)=\max(0,xW_1+b_1)W_2+b_2 \quad (8)$$

Here, W_1 and W_2 are the weights of the network, and b_1 and b_2 are the bias parameters of the network. Finally, the output of each sub-layer is added with the residual of the input and layer normalized. When traditional LST models deal with dynamic NTDs, they often need to retrain the entire model to adapt to new data. Therefore, this study introduces an incremental update mechanism so that the model can dynamically adapt to information data without retraining the entire model. The first step is sliding window maintenance, that is, maintaining a fixed-size sliding window to store the latest NTD [23]. When new data arrive, the most recent data are removed from the window and the new one is added to the window. The study set the sliding window size as an empirical value (1000), which corresponds to approximately 15–30 minutes of normal network traffic data volume. The selection of this size is based on a trade-off: a smaller window (<500) can respond more quickly to the latest changes, but may result in unstable detection of periodic or persistent attacks due to insufficient retained historical information; a larger window (>2000) helps stabilize the model, but reduces the response speed to sudden new threats and increases computational latency. The final step is to update only the parts of the model that are relevant to the new data, as shown in formula (9):

$$\varsigma_n=\varsigma_o+\eta\cdot\nabla_{\varsigma}L(x_n,\varsigma_o) \quad (9)$$

Here, ς_n and ς_o are the updated model and old model parameters, and η is the learning rate. The traditional LST model has high computational complexity and limited ability to capture local features when processing high-resolution NIdata. Therefore, this study introduces atrous (dilated) convolution to improve the LST model. Atrous convolution is a technology that expands the receptive field of the convolution (conv) kernel. By introducing intervals in the conv kernel, the receptive field of the conv kernel can be expanded without increasing computational complexity. Spatial convolution enables the model to capture local features more effectively while reducing the amount of computation. The framework of atrous convolution is shown in Figure 4.

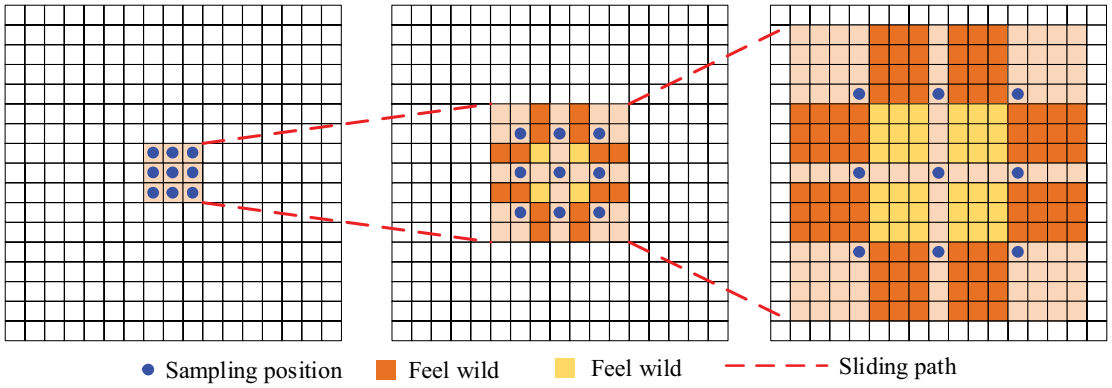


Figure 4. Atrous convolution.

The atrous convolution calculation in Figure 4 is shown in formula (10):

$$b[i]=\sum_{k=0}^{K-1}\omega[k]\cdot a[i+k\cdot\vartheta] \tag{10}$$

Here, b and a are the output and input of atrous convolution, ω and K are the weight and size of the conv kernel, and d is the hole rate. The overall process based on RTDNI is shown in Figure 5.

In Figure 5, the NTD simulated by the LDMs model is preprocessed. The preprocessed NTD is put into the LST model, and position coding is added to each data point. A multi-head SAM is used to capture long-distance dependencies in data. The FNN in each Transformer layer per-

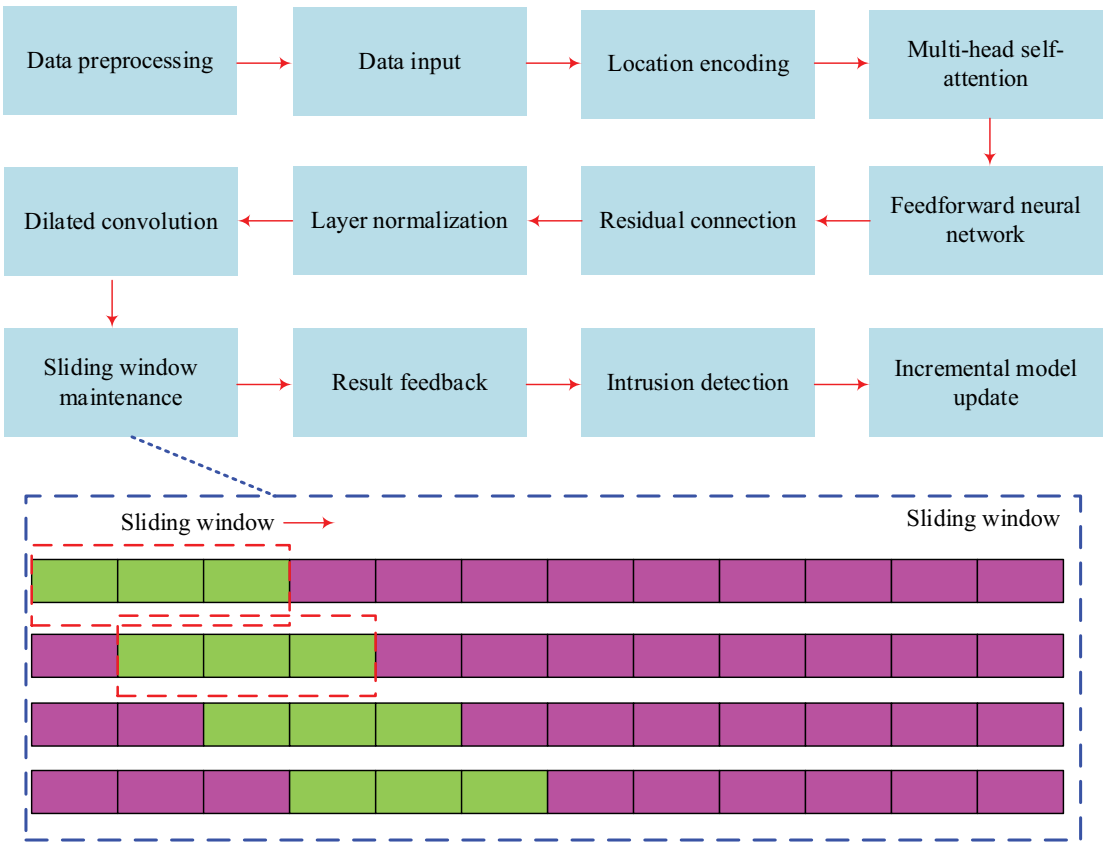


Figure 5. The overall detection process for network intrusions.

forms nonlinear transformation on the processed data. The output of the FNN is residually connected to the input data, and layer normalization is performed. In the LST, an atrousconv layer is inserted, and the receptive field of the conv kernel is expanded through atrous convolution. A fixed-size sliding window is maintained to store the most recent NTD. Given the data in the sliding window, the parts related to the new data are updated, and the updated model is used to detect intrusions in real-time NTDs, and then the output of the model is used to determine whether there is an intrusion. Finally, the detection results are fed back, and the model parameters are adjusted and optimized based on the actual detection results to further improve the performance and adaptability of NID.

4. Results

4.1. Performance Verification of the LST Model

To test the performance of the proposed LST model, this study conducts a series of performance verification experiments in a high-performance experimental environment. Table 1 shows the configuration of the experimental environment and the LST model.

The datasets used are KDD CUP 99 and NSL-KDD, both of which are free and open source. The KDD Cup 99 dataset has 311,029 data points, and the main categories are divided into

Table 1. Parameter settings for the environment and LST model.

Component	Model/Specification	Parameter	Value
CPU	Intel Core i9-12900K @ 3.20 GHz, 16 Cores, 24 Threads	Model type	LST
GPU	NVIDIA GeForce RTX 3090, 24 GB GDDR6X	Number of layers	6
RAM	64 GB DDR4 @ 3200 MHz	Hidden dimension	512
Storage	1 TB NVMe SSD (Primary) + 4 TB HDD (Secondary)	Number of attention heads	8
Operating System	Ubuntu 20.04 LTS	Feed-forward dimension	2048
Python	3.8.10	Dropout rate	0.1
CUDA	11.4	Positional encoding	Sine/cosine encoding
cuDNN	8.2.4	Dilation rate for atrous convolution	2
TensorFlow	2.6.0	Batch size	32
Network Interface	10 GbE Ethernet Card	Learning rate	0.0001
/	/	Optimizer	Adam
/	/	Training epochs	200

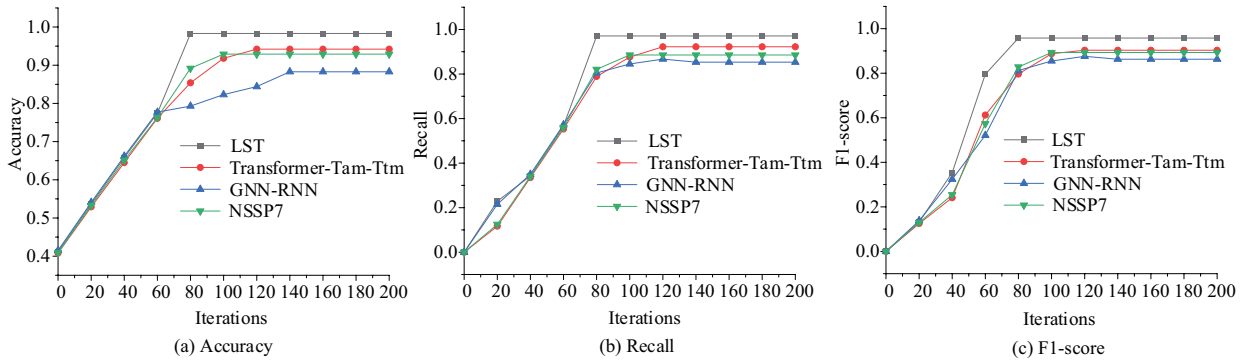


Figure 6. Comparison of accuracy rate, recall rate, and F1-score curves of the models

normal, Denial-of-Service attack (DoS), User to Root (U2R), Remote to Local (R2L) and Probe attack (Probe). NSL-KDD has 125,973 data points in the same categories as KDD Cup 99. To evaluate the generalization ability, this paper adopts 5-fold cross-validation, that is, the original dataset is randomly divided into 5 mutually exclusive subsets of approximately equal sizes. Each subset contains a training set and a test set. In each round of verification, one subset is selected as the test set, and the remaining 4 subsets are the training set. The dataset is segmented into a training set, a verification set, and a test set in a ratio of 7:2:1. This study also selects Transformer-Tam-Ttm, NSSP7, and GNN-RNN models for control experiments. The comparison of accuracy, recall, and F1-score curves based on the three models is shown in Figure 6.

Figure 6 shows that the LST model not only has a fast convergence speed, but also has better performance indicators than other models after convergence. Specifically, LST converges after 80 iterations, and its precision, recall, and F1-scores tend to 0.983, 0.971, and 0.958, respectively. The performance slightly worse than LST is Transformer-Tam-Ttm, whose three convergence indicators tend to 0.942, 0.922, and 0.903 after 120 iterations. After GNN-RNN converges, its performance is the worst, and the convergence speed is also the slowest. It converges after 140 iterations, and the three values tend to 0.883, 0.852, and 0.863. This study further explores the generalization ability of each model through 5-fold cross-validation, as shown in Figure 7.

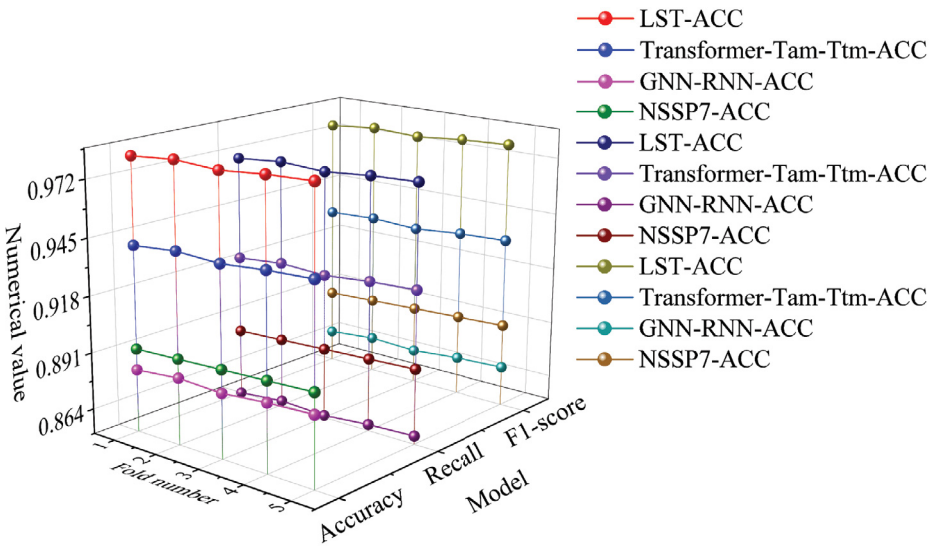
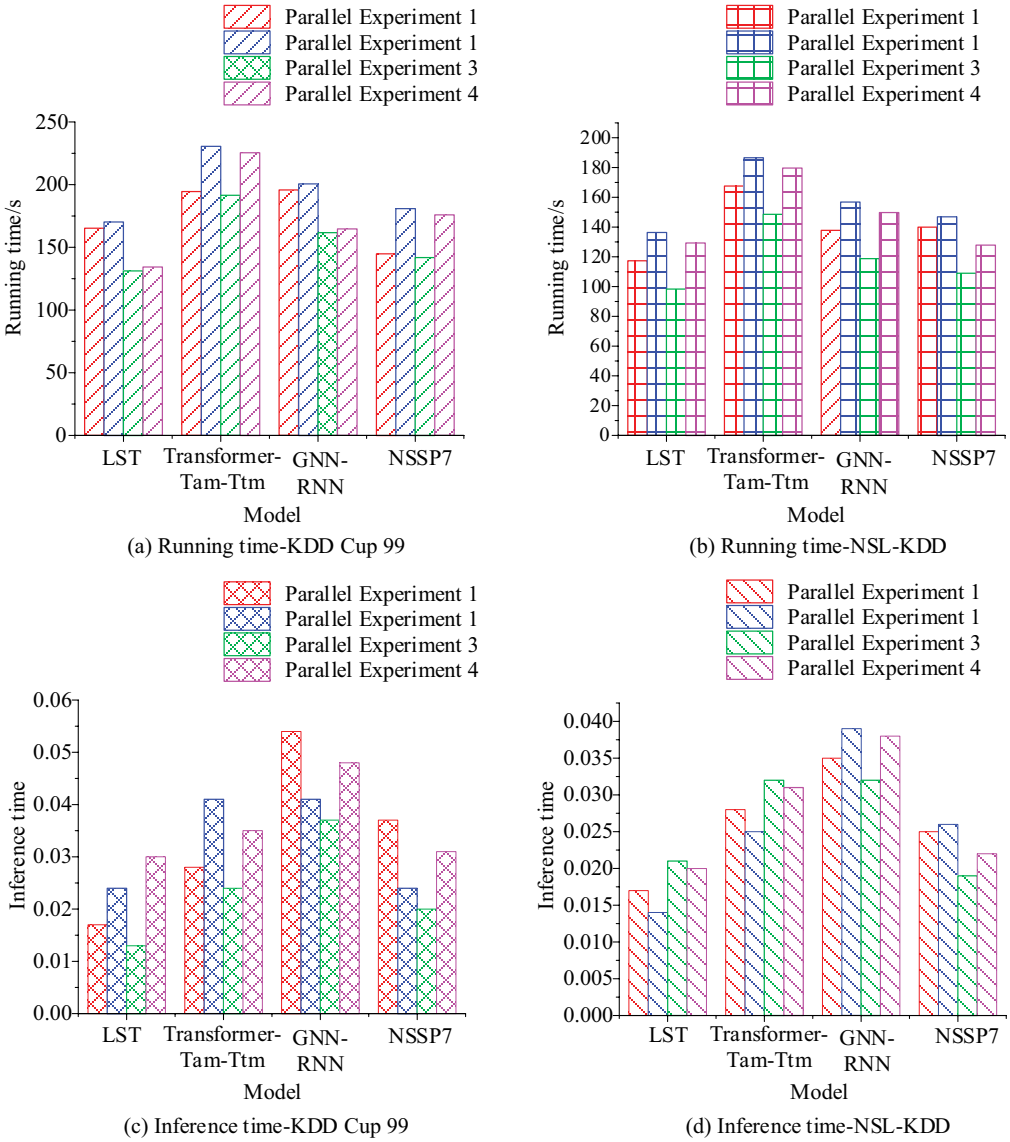


Figure 7. Five-fold cross-validation results for each model.

In Figure 7, LST shows excellent performance in ACC, recall, and F1-score. Its accuracy rate is between 0.982 and 0.986, its recall rate is between 0.970 and 0.974, and its F1-score is between 0.976 and 0.980, showing extremely high stability and excellent comprehensive performance. Although the overall performance of Transformer-Tam-Ttm is slightly inferior to LST, it also shows good stability. Its accuracy value tends to 0.941–0.945, the recall rate tends to 0.921–0.925, and the F1-score tends to 0.931–0.934. The performance of GNN-RNN is relatively low, but the results still have a certain stability, with accuracy, recall, and F1-scores ranging from 0.882 to 0.886, 0.852 to 0.856, and 0.867 to 0.871. The performance of NSSP7 is between Transformer-Tam-Ttm and GNN-RNN, with three metrics ranging from 0.892 to 0.896, 0.884 to 0.888, and 0.888 to 0.892. In summary, the LST model performs the best, and although the other three models have differences in performance, they all show a certain degree of stability and effectiveness in the 5-fold cross-validation. This study further compares the running time and inference time of the four models, as shown in Figure 8.

Figures 8 (a) and (b) show the running time comparison of the four models under the KDD CUP 99 and NSL-KDD datasets. LST shows significant efficiency advantages on both datasets. On KDD CUP 99, the average running time of LST is 150.254 s, which is sig-



(a) Running time-KDD Cup 99

(b) Running time-NSL-KDD

(c) Inference time-KDD Cup 99

(d) Inference time-NSL-KDD

Figure 8. Comparison of running and inference time among models.

nificantly shorter than other models. Transformer-Tam-Ttm has the longest running time, reaching 210.496 s, while GNN-RNN is 180.648 s and NSSP7 is 160.892 s. On NSL-KDD, the running time of LST is further shortened to 120.335 s, Transformer-Tam-Ttm is 170.567 s, GNN-RNN is 140.782 s, and NSSP7 is 130.901 s. The LST model can complete training tasks more efficiently when processing large-scale datasets, significantly reducing the time required for model training. Figures 8 (c) and (d) show the comparison of inference time of the four models. LST also performs well, with average inference times of 0.021 s and 0.018 s, values significantly lower than Transformer-Tam-Ttm (0.032 s and 0.029 s), GNN-RNN (0.045 s and 0.036 s), and NSSP7 (0.0.28 s and 0.023 s). This shows that LST has excellent real-time performance in practical applications, and it can respond quickly to NTDs and detect potential intrusions in a timely manner. Finally, to clearly demonstrate the superiority of the LST model, the study conducted a comparative analysis of it with SMOTE, CGAN, and data replay. The experiments were carried out on the CIC-IDS2018 dataset, which covers 14 types of attacks, including traditional attacks (such as DoS/DDoS) and advanced attacks (such as penetration, botnets, web attacks, *etc.*). The results are shown in Table 2.

In Table 2, in terms of generation quality, the Fréchet distance of LDM was only 15.3, significantly lower than that of CGAN (31.8). In the downstream detection task, LDM achieved a macro-average F1 score of 0.951, which was 2.9% higher than the best competing method CGAN, and the 95% confidence interval did not overlap. The paired t-test showed that the difference was highly statistically significant ($p < 0.01$). Although the training time of CLDM (285.4s) was longer than that of the traditional over-sampling method, through adaptive noise scheduling optimization, its efficiency has approached the level of CGAN, and the performance improvement it brings is significant, verifying the good balance between quality and efficiency of this method. To quantify the data

Table 2. Performance comparison results of the considered models.

Method	SWRL rule	Rule description	Context dependency	F1-score	Training time / s
SMOTE	/	0.912 ± 0.009^{d}	0.873 ± 0.013^{c}	0.892 ± 0.010^{d}	45.2 ± 2.3^{d}
	/	(0.903, 0.921)	(0.860, 0.886)	(0.882, 0.902)	
CGAN	31.8 ± 1.6	0.935 ± 0.007^{b}	0.910 ± 0.011^{b}	0.922 ± 0.008^{c}	312.5 ± 15.7^{a}
	(30.2, 33.4)	(0.928, 0.942)	(0.899, 0.921)	(0.914, 0.930)	
Data Replay	/	0.926 ± 0.008^{c}	0.902 ± 0.012^{b}	0.914 ± 0.009^{b}	88.6 ± 4.5^{c}
	/	(0.918, 0.934)	(0.890, 0.914)	(0.905, 0.923)	
LDM	15.3 ± 0.9	0.960 ± 0.005^{a}	0.943 ± 0.007^{a}	0.951 ± 0.006^{a}	285.4 ± 14.2^{b}
	(14.4, 16.2)	(0.955, 0.965)	(0.936, 0.950)	(0.945, 0.957)	

Table 3. Data quantification assessment results.

Method	Data generation precision	Data generation recall	C2ST accuracy	Feature variance ratio
SMOTE	0.85 ± 0.02	0.40 ± 0.03	0.92 ± 0.02	0.68 ± 0.05
CGAN	0.88 ± 0.02	0.61 ± 0.03	0.73 ± 0.03	0.82 ± 0.04
Data Replay	0.88 ± 0.02	0.52 ± 0.03	0.89 ± 0.02	0.75 ± 0.04
LDM	0.94 ± 0.01	0.87 ± 0.02	0.58 ± 0.02	0.96 ± 0.02

quality and authenticity, the study further analyzed the accuracy of the generated data, the recall rate of the generated data, the C2ST accuracy rate, and the feature variance ratio. The results are shown in Table 3.

In Table 3, regarding the balance between quality and diversity, LDM achieved the highest generation accuracy (0.94) and recall rate (0.87), demonstrating that the generated samples are not only highly realistic but also can fully cover the distribution range of the real data. Its C2ST accuracy rate (0.58) is closest to the random guessing level, indicating that the generated data is almost indistinguishable from the real data, and has the strongest realism. Additionally, the feature variance ratio of LDM (0.96) is closest to 1, indicating that the generated data has completely captured the feature variations of each dimension of the real data. To directly verify the causal relationship between the simulated data and the improvement in detection performance, the study designed an ablation experiment for the system. The experiment fixed the use of the improved LST model architecture and hyperparameters, and only changed the composition of the training data. It was compared with the other models on the CIC-IDS2018 dataset. The ablation experiment results are shown in Table 4.

Table 4. The results of the ablation experiment.

Data settings	Average F1 score of the macro	Average detection rate of minority class attacks	Model stability
Baseline-Real	0.874 ± 0.012	0.723	0.012
Baseline-Real+SMOTE	0.892 ± 0.010	0.768	0.010
Baseline-Real+CGAN	0.922 ± 0.008	0.841	0.008
Ablation-LDM Only	0.913 ± 0.009	0.826	0.009
Proposed-Full	0.951 ± 0.006	0.928	0.006

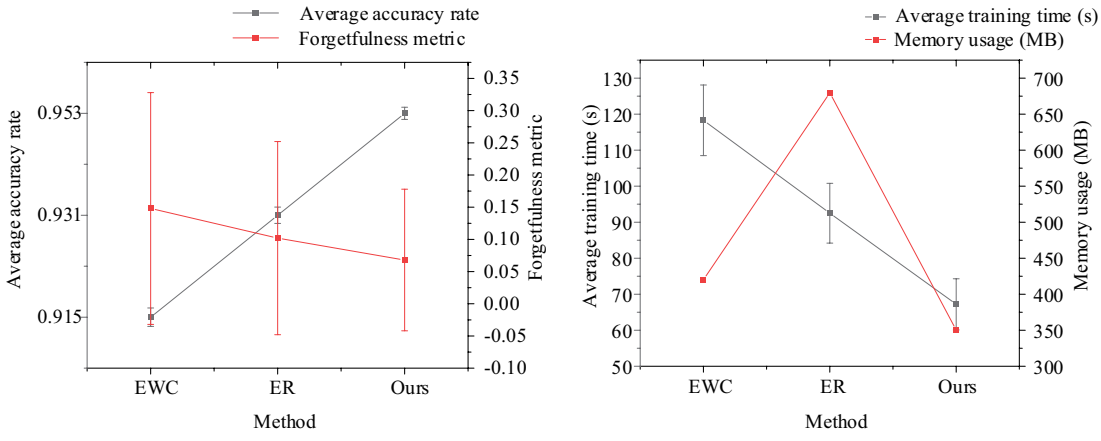
In Table 4, the model trained solely using the data generated by LDM (Ablation-LDM Only) achieved a F1 score of 0.913, significantly higher than the baseline using only the original data (0.874), demonstrating that the simulated data itself has independent teaching value and can train an effective detection model without the need for real data. The complete method (Proposed-Full) achieved the best performance and highest stability, indicating that real data and high-quality simulated data have a synergistic enhancement effect. Moreover, compared with the baselines that included SMOTE or CGAN data, the addition of LDM data brought a more significant performance improvement, directly proving that the superiority of the quality and diversity of the LDM-generated data is the key reason for the improvement in detection performance. The study also included a comparative experiment with two classic incremental/sustained learning methods. The experiment was conducted on the CIC-IDS2018 dataset, simulating 5 consecutive tasks, each containing different combinations of attack types. The subsequent tasks introduced new attacks to simulate the real incremental environment. The results are shown in Figure 9.

In Figure 9, the research method demonstrates superior performance, anti-forgetting ability, and efficiency. In terms of average accuracy, the method proposed in this paper (0.953) significantly outperforms EWC (0.915) and ER (0.931). In terms of anti-forgetting ability, the forgetting metric of this method is the lowest (0.068), indicating that it can more stably retain historical knowledge. Moreover, the research method outperforms the baseline in terms of training time (67.2 s) and memory usage (350 MB), thanks to the sliding window mechanism and local parameter update strategy, which avoids the regularization computation cost of EWC and the storage cost of ER. These results demonstrate the comprehensive advantages of the incremental strategy proposed in this study in the real-time network intrusion detection scenario.

4.2. RTDNI Effect Analysis

After verifying the performance of the LST model, this study further analyzes the model's detection rate for different network intrusions. The comparison of detection rates based on the four models is shown in Figure 10.

Figures 10 (a)~(d) shows the comparison of the detection rate curves of the model for DoS, U2R, R2L, and Probe intrusions. LST shows the highest numerical level of detection rate for different network intrusion methods. The detection rate of LST converges to 98.9%, 99.2%, 98.1%, and 99.5%, respectively, for each considered intrusion. In comparison, the detection rate of Transformer-Tam-Ttm is slightly lower than that of LST, ranging from 94% to 98%. GNN-RNN has the worst detection rate, with values ranging from 88% to 92%, especially



(a) Average accuracy rate and forgetting metric

(a) Average training time and memory usage for tasks

Figure 9. Performance comparison results of the proposed model with two classic incremental learning methods.

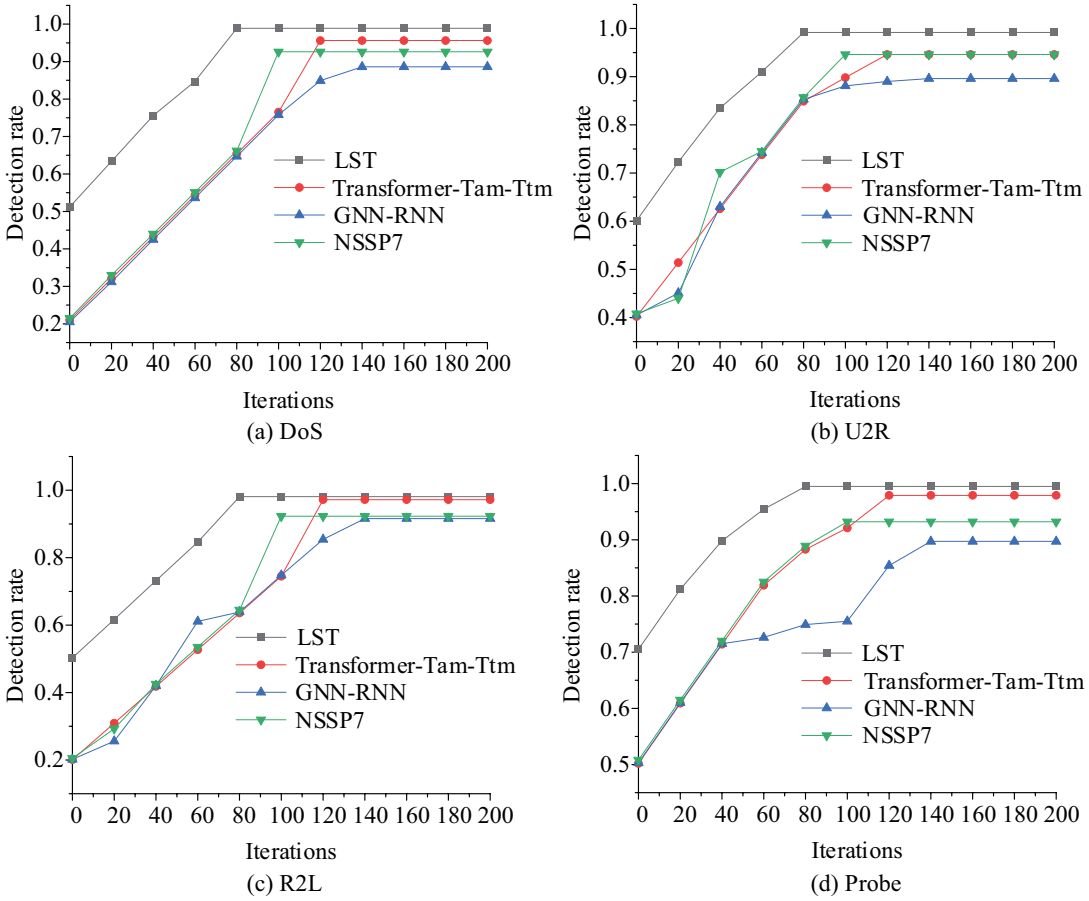


Figure 10. Comparison of the detection rates.

only 88.6% in DoS detection. The detection rate of NSSP7 is between Transformer-Tam-Ttm and GNN-RNN, and its value is between 92% and 95%. This study further compares the resource consumption levels for different network intrusion methods, as shown in Figure 11.

In Figure 11, LST shows excellent performance. In terms of CPU usage, LST is only 45.2% when dealing with DoS attacks and only 48.5% in Probe detection, indicating that its demand for CPU resources during runtime is relatively low and it can utilize computing resources more efficiently. In terms of memory usage, LST also shows low occupancy, such as only 12.5 GB in U2R attack detection, which helps run in resource-constrained environments. Transformer-Tam-Ttm has relatively high resource consumption, especially in terms of memory usage, such as reaching 16.0 GB in Probe detection. The resource consumption of GNN-RNN and NSSP7 is between LST and Transformer-Tam-Ttm, but also shows higher resource requirements in some cases, such as GNN-RNN's CPU usage of 51.4% in R2L attack detection. In summary, LST shows a better balance in resource consumption. It can rationally utilize computing resources while maintaining high detection performance, making LST more suitable for real-time NID systems.

To more comprehensively verify the model's performance in actual NIDs, this study uses a self-made real-world dataset. This dataset is generated by simulating a complex network environment and covers a variety of network intrusion behaviors, including two more complex network intrusion methods: Advanced Persistent Threat (APT) and Zero-Day attack. A zero-day attack refers to a sudden network attack launched by an attacker against a target system, exploiting a security vulnerability that has not yet been discovered by the software vendor or has been discovered but no patch has been released yet. APT refers to a complex, persistent, and usually initiated by attackers

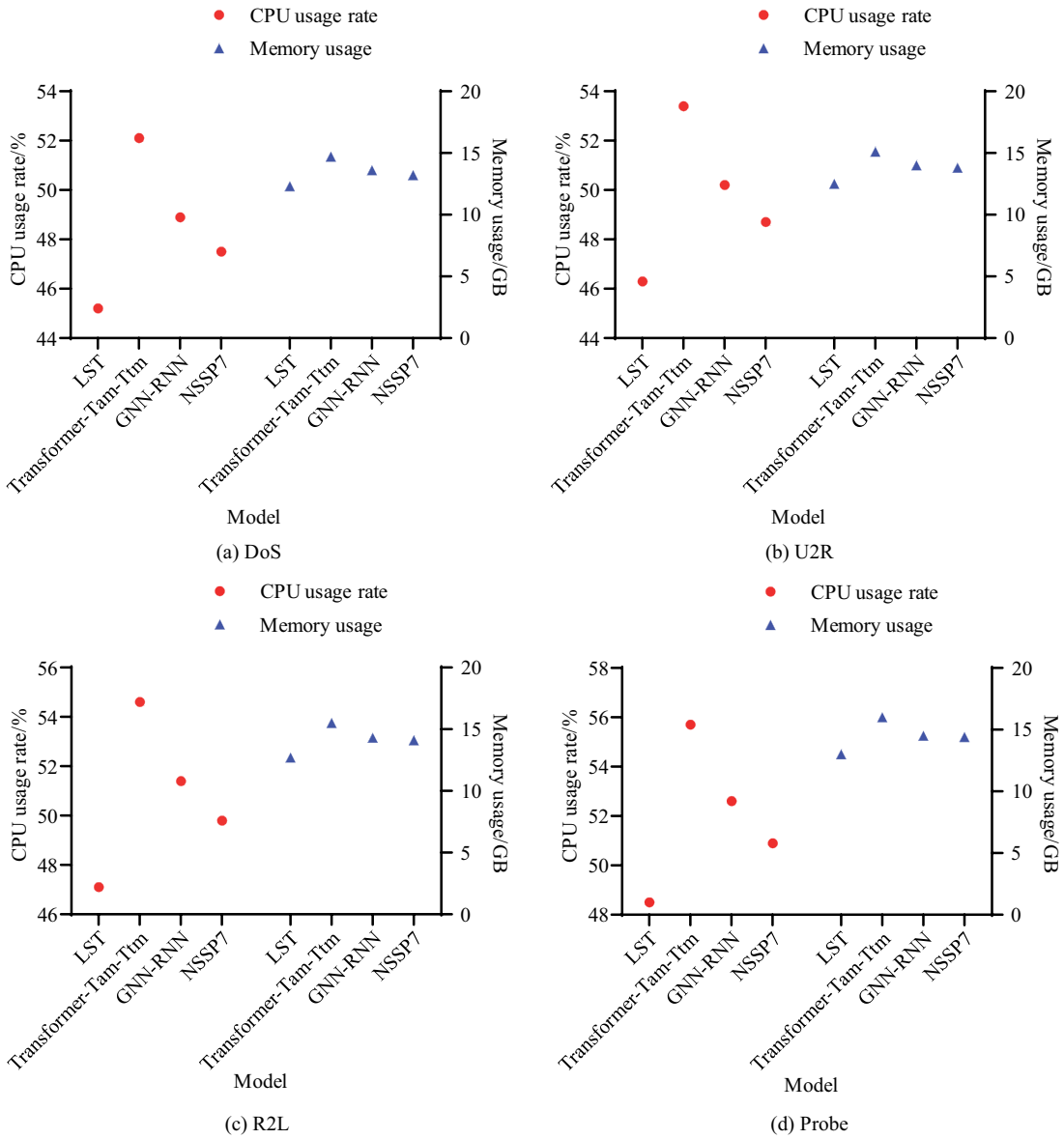


Figure 11. Comparison of resource consumption levels.

with abundant resources as a type of cyber intrusion activity. In addition, this dataset simulates an enterprise network consisting of three subnets, covering web servers, database servers, internal office terminals, firewalls, routers and other network devices. At the same time, approximately 500,000 network flow records were generated, including normal traffic (70%), traditional known attacks (25%), APT and zero-day attacks (5%). The comparison of detection rates and running times of the four models for complex network intrusion methods is listed in Table 5.

In Table 5, LST shows excellent performance. For APT attacks, the detection rate of LST is as high as 97.4%, which is significantly better than Transformer-Tam-Ttm (95.2%), GNN-RNN (92.8%) and NSSP7 (94.5%). In terms of zero-day attack detection, LST also outperforms the control model with a detection rate of 96.8%. In terms of running time, LST also shows high efficiency. Its running time for handling APT and zero-day attacks is 160.3 s and 155.2 s, which are shorter than other models. LST not only has high accuracy in detecting complex network intrusion behaviors, but also has obvious advantages in operational efficien-

Table 5. Detection rate and running time of considered network intrusion methods.

Network intrusion methods	Model	Detection rate/%	Running time/s
APT	LST	97.4	160.3
	Transform-er-Tam-Ttm	95.2	200.5
	GNN-RNN	92.8	180.7
	NSSP7	94.5	170.9
Zero-Day attack	LST	96.8	155.2
	Transform-er-Tam-Ttm	94.3	195.4
	GNN-RNN	91.5	175.6
	NSSP7	93.7	165.8

cy, giving it a significant advantage when dealing with complex NID tasks in the real world. The study further compared the complexity and computational cost of the four models, and the results are shown in Table 6.

In Table 6, the LST model achieves an optimal balance between computational efficiency and detection performance. Although its parameter size (20.4M) and computational cost (3.8 G FLOPs) are not the lowest, its training time (285.4 s) is significantly shorter than that of the Transformer-Tam-Ttm model with the largest parameter size (312.5s), and its single-sample inference time (18 ms) is the shortest among the four models, being less than half of that of GNN-RNN (36 ms). This indicates that LST, by integrating atrous convolution and incremental update designs, achieves higher computational and inference efficiency with a moderate model complexity.

Table 6. Comparison of model complexity and computational cost.

Model	Parameter quantity (millions)	FLOPs(G)	Training time /s	Inference time / ms	Memory usage / MB
LST	20.4	3.8	285.4 ± 14.2	18.0 ± 1.5	520
Transformer-Tam-Ttm	48.7	7.2	312.5 ± 15.7	29.0 ± 2.1	810
GNN-RNN	35.2	5.4	180.7 ± 9.8	36.0 ± 2.8	650
NSSP7	12.8	2.1	170.9 ± 8.5	23.0 ± 1.9	380

4. Discussion and Interpretation

To improve the precision of NID, this study proposed an incremental RTDNI method based on LDMs and LST models. In the experiment, LST converged after 80 iterations, and its convergence speed was significantly better than the control models. Meanwhile, its accuracy, recall rate, and F1-score tended to 0.983, 0.971, and 0.958, respectively, and its performance indicators were also better than other models. In addition, the average running time of LST was only 150.254 s, and the inference time was only 0.021 s, indicating that it has excellent real-time performance and can respond quickly to NTD. Ma *et al.* proposed an improved LST model based on Gaussian attention [24] in view of the current situation that CNN is difficult to model long-distance dependencies and has high computational overhead in hyperspectral image classification. This method was better than the best existing model on four public datasets, and the accuracy was improved by 0.62% compared to the baseline model. It took into account the advantages of global feature extraction and computational efficiency.

All the experimental results of the study (including means, standard deviations and confidence intervals) were obtained under the condition of the fixed random seed. Although the fixed seed ensures the complete reproducibility of a single experimental sequence, the study evaluated the stability of the results by running multiple independent experiments (each run changed the global seed, but the internal random process was still controlled) and reported their statistics. This ensured that the observed performance advantages were a stable reflection of the model's capabilities, rather than an accidental result of a single random run.

In terms of RTDNI, LST also performed better. Specific LST targeted different network intrusion methods, and its detection rate was above 98%. In the self-made real-life dataset, LST also showed good real-time performance and high accuracy. Its detection rate reached 97.4% when detecting APT attacks, and 96.8% when detecting zero-day attacks. The running time was 160.3 s and 155.2 s, and the values were significantly better than the control models. Ullah *et al.* proposed an imbalanced network traffic intrusion detection system based on Transformer migration learning [25] to address the current situation where NTD imbalance in NID makes it difficult to identify specific attacks. The method performed well in detailed experiments on UNSW-NB15, CIC-IDS2017, and NSL-KDD datasets, and was able to effectively learn network feature representation and detect a small number of attack types, while improving the credibility of the model through explainable artificial intelligence methods. The difference is that although the model mentioned in the report and the research model are both variants of the Transformer model, there are certain differences in the definitions of network intrusion locations. The attention matrix of the research model comes from the coordinates to be estimated and the known anchor points, and the attention weight is interpreted as the similarity of the positioning fingerprint. In the understanding of position in this report, it is only to preserve the order, and there is less attention to absolute position, which can be replaced by relative position encoding.

In summary, the proposed research method has significant advantages in RTDNI and is suitable for real-time monitoring tasks in complex network environments, with excellent accuracy and efficiency.

5. Conclusion

In response to the increasingly complex network environment and evolving network intrusion threats, this study proposes an incremental RTDNI algorithm optimization method based on computer data simulation. By combining LDMs and LST models, the performance and adaptability of the NID system are effectively improved. Research shows that the proposed model exhibits high accuracy and efficiency in detecting complex network intrusion behaviors, significantly improving real-time monitoring capabilities. However, this study still has certain shortcomings, and the resource consumption of the model is still high when processing large-scale data sets, especially in terms of memory usage. Future work will further optimize the model architecture to reduce resource consumption and improve the interpretability of the model.

Declaration of Competing Interests

The authors declare no competing interests.

Funding

None declared.

Data Availability

Data used in this article are available upon request from the authors.

References

[1] D. H. De, B. T. Khoa, V. T. T. Nguyen, "Customer's Online Purchase Intention: The Role of Perceived Business Size and Reputation", *Journal of Logistics, Informatics and Service Science*, vol. 10, no. 2, pp. 296–307, 2023. http://dx.doi.org/10.33168/JLISS.2023.0304

[2] L. N. Han, X. Z. Ma, J. D. Tan, *et al.,* "Balancing material supply-demand with arima and neural networks", *International Journal of Simulation Modelling*, vol. 22, no. 4, pp. 712–722, 2023. http://dx.doi.org/10.2507/IJSIMM22-4-C018

[3] J. Alshemmari, "An empirical study on employee empowerment role in increasing efficiency of employee performance", *Journal of Logistics, Informatics and Service Science*, vol. 10, no. 1, pp. 52–71, 2023. http://dx.doi.org/10.33168/JLISS.2023.0104

[4] I. C. Rădulescu, "Robust Model Predictive Control for Systems Affected by Constant and Norm 2 Bounded Disturbances", *Studies in Informatics and Control*, vol. 33, no. 1, pp. 99–106, 2024. http://dx.doi.org/10.24846/v33i1y202409

[5] Y. Li, Y. Xiao, W. Liang, *et al.,* "The security and privacy challenges toward cybersecurity of 6G networks: A comprehensive review", *Computer Science and Information Systems*, vol. 21, no. 3, pp. 851–897, 2024. http://dx.doi.org/10.2298/csis230804016l

[6] J. Sheng, "QoS Technology in Computer Communication Transmission Network Security under the Artificial Intelligence", *Tehnički vjesnik*, vol. 31, no. 6, pp. 1938–1949, 2024. http://dx.doi.org/10.17559/tv-20231023001049

[7] S. Kara, S. Hizal, A. Zengin, "Design and implementation of a DEVS-based cyber-attack simulator for cyber security", *International Journal of Simulation Modelling*, vol. 21, no. 1, pp. 53–64., 2022. http://dx.doi.org/10.2507/JSIMM21-1-587

[8] D. Breskuvienė, G. Dzemyda, "Categorical feature encoding techniques for improved classifier performance when dealing with imbalanced data of fraudulent transactions", *International Journal of Computers Communications & Control*, vol. 18, no. 3, pp. 5433, 2023. http://dx.doi.org/10.15837/ijccc.2023.3.5433

[9] S. G. AboulEla, R. F. Kashef, "Leveraging large language models, graph neural networks, and explainable AI for revolutionizing the next-generation network intrusion detection systems", *Journal of Intelligent Information Systems*, vol. 63, no. 5, pp. 1807–1835, 2025. http://dx.doi.org/10.1007/s10844-025-00964-2

[10] M. S. Alshehri, O. Saidani, F. S. Alrayes, *et al.*, "A self-attention-based deep convolutional neural networks for IIoT networks intrusion detection, *IEEE Access*, vol. 12., no. 2, pp. 45762–45772, 2024. http://dx.doi.org/10.1109/ACCESS.2024.3380816

[11] Y. Imrana, Y. Xiang, L. Ali, *et al.,* "CNN-GRU-FF: a double-layer feature fusion-based network intrusion detection system using convolutional neural network and gated recurrent units", *Complex & Intelligent Systems*, vol. 10, no. 3, pp. 3353–3370, 2024. http://dx.doi.org/10.1007/s40747-023-01313-y

[12] J. He, X. Wang, Y. Song, *et al.,* "Network intrusion detection based on conditional wasserstein variational autoencoder with generative adversarial network and one-dimensional convolutional neural networks", *Applied Intelligence*, vol. 53, no. 10, pp. 12416–12436, 2023. http://dx.doi.org/10.1007/s10489-022-03995-2

[13] J. Ribeiro, R. Santos, C. Analide, *et al.,* "Implementing Federated Learning and Explainability Techniques in Regression Models to Increase Transparency and Reliability", *Studies in Informatics and Control*, vol. 33, no. 4, pp. 15–24, 2024. http://dx.doi.org/10.24846/v33i4y202402

[14] W. Xie, J. He, F. Huang, *et al.,* "Supply chain financial fraud detection based on graph neural network and knowledge graph", *Tehnički vjesnik*, vol. 31, no. 6, pp. 2055–2063, 2024. http://dx.doi.org/10.17559/TV-20240606001759

[15] R. Bubliene, E. Trinkuniene, "Protection Against Multiple Discrimination in EU Member States Law: A Case Study", vol. 4, no. 1, pp. 21–31, 2023. http://dx.doi.org/10.33168/SISD.2023.0103

[16] T. Laham, F. Sherbaji, S. Mouselli, "Pain or Gain? The Impact of Sanctions on the Sustainability of Banks' Services", *Journal of Service, Innovation and Sustainable Development*, vol. 4, no. 1, pp. 87–99, 2023. http://dx.doi.org/10.33168/SISD.2023.0108

[17] M. K. Bhuyan, M. Kamruzzaman, S. I. Nilima, *et al.,* "Convolutional Neural Networks Based Detection System for Cyber-Attacks in Industrial Control Systems", *Journal of Computer Science and Technology Studies*, vol. 6, no. 3, pp. 86–96, 2024. http://dx.doi.org/10.32996/jcsts.2024.6.3.9

[18] J. Merkevičius, J. Nalivaikė, D. Nalivaika, "Expanding Internet of Things into the New Markets", *Journal of management changes in the digital era.*, vol. 1, no. 1, pp. 42–59, 2024. http://dx.doi.org/10.33168/JMCDE.2024.0104

[19] S. Dalal, P. Manoharan, U. K. Lilhore, *et al.,* "Extremely boosted neural network for more accurate multi-stage Cyber attack prediction in cloud computing environment", *Journal of Cloud Computing*, vol. 12, no. 1, pp. 1–22, 2023. http://dx.doi.org/10.1186/s13677-022-00356-9

[20] B. Ghimire, "Determinants of Entrepreneurial Readiness: An Empirical Analysis", *Journal of Management Changes in the Digital Era*, vol. 1, no. 1, pp. 145–155, 2024. http://dx.doi.org/10.33168/JMCDE.2024.0110

[21] J. Bi, K. Xu, H. Yuan, *et al.,* "Network attack prediction with hybrid temporal convolutional network and bidirectional GRU", *IEEE Internet of Things Journal*, vol. 11, no. 7, pp. 12619–12630, 2023. http://dx.doi.org/10.1109/JIOT.2023.3334912

[22] A. I. Jony, A. K. B. Arnob, "A long short-term memory based approach for detecting cyber attacks in IoT using CIC-IoT2023 dataset", *Journal of edge computing*, vol. 3, no. 1, pp. 28–42, 2024. http://dx.doi.org/10.55056/jec.648

[23] T. Tabassum, O. Toker, M. R. Khalghani, "Cyber-physical anomaly detection for inverter-based microgrid using autoencoder neural network", *Applied Energy*, vol. 355, no. 1, pp. 122283–122298, 2024. http://dx.doi.org/10.1016/j.apenergy.2023.122283

[24] C. Ma, M. Wan, J. Wu, *et al.,* "Light self-Gaussian-attention vision transformer for hyperspectral image classification", *IEEE transactions on instrumentation and measurement*, vol. 72, no. 2, pp. 1–12, 2023. http://dx.doi.org/10.1109/TIM.2023.3279922

[25] F. Ullah, S. Ullah, G. Srivastava, *et al.,* "IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic", *Digital Communications and Networks*, vol. 10, no. 1, pp. 190–204, 2024. http://dx.doi.org/10.1016/j.dcan.2023.03.008

Received: February 2026
Revised: March 2026
Accepted: March 2026

Contact addresses:
Hui Xiong
Sichuan Vocational and Technical College
Suining City
Sichuan Province
China
e-mail: xionghui811129@163.com

Chao Yuan
Sichuan Vocational and Technical College
Suining City
Sichuan Province
China
e-mail: yuanchao_scvtc1983@163.com

Ping Zhao
Sichuan Vocational and Technical College
Suining City
Sichuan Province
China
e-mail: zhaoping667788@163.com

Hui Xiong obtained her M.Sc. degree in computer application technology from The XinJiang Technical Institute of Physics & Chemistry, China in July 2007. She is currently a lecturer at Sichuan Vocational and Technical College, Suining, Sichuan, China. Her research interests include software engineering and artificial intelligence.

Chao Yuan obtained her B.Sc. degree in computer science and technology from the Sichuan Normal University, China in July 2007. She is currently associate professor at Sichuan Vocational and Technical College, Suining, Sichuan, China. Her research interests include networking and information security.

Ping Zhao obtained her B.Eng. degree in computer science and technology from the Mianyang Teachers' College, China in June 2009. She is currently a lecturer at Sichuan Vocational and Technical College, Suining, Sichuan, China. Her research interests include artificial intelligence and big data.