scientific reports

Check for updates

OPEN

An attack detection method based on deep learning for internet of things

Yihan Yu, Yu Fu, Taotao Liu✉, Kun Wang & Yishuai An

With the rapid development of Internet of Things (IoT) technology, the number of network attack methods it faces is also increasing, and the malicious network traffic generated is growing exponentially. To identify attack traffic for the protection of IoT device security, attack detection has attracted widespread attention from researchers. However, current attack detection methods struggle to identify complex and variable attack methods, resulting in a high false positive rate. Additionally, feature redundancy and class imbalance in IoT traffic datasets also constrain detection performance. To address these issues, this paper proposes an attack detection method based on deep learning for IoT. Firstly, a genetic algorithm is used for feature selection; secondly, a cost-sensitive function is employed to address the scarcity of attack traffic in IoT; and finally, a combination of Convolutional Neural Networks and Long Short Term Memory Network is utilized to extract spatiotemporal information from the network. The results demonstrate that this method exhibits superior performance on two IoT benchmark datasets, effectively enhancing the performance of IoT attack detection.

Keywords Internet of things, Attack detection, Feature selection, Class imbalance

With the advancement of 5G mobile communication technology, the Internet of Things (IoT) is rapidly emerging, interconnecting billions of physical objects—including sensors, smart vehicles, and other electronic/software-embedded devices—to enable data exchange[1]. This connectivity facilitates advanced IoT applications and extends automation to daily life[2]. However, massive device deployments generate enormous traffic volumes, attracting increasing numbers of malicious attackers. Consequently, attacks targeting IoT devices have surged dramatically. A smart home hacking survey revealed that devices such as TVs, thermostats, smart kettles, and security systems suffered over 12,000 cyberattacks from criminals and unknown entities within a single week[3]. This imposes significant pressure on data communication, resource management, Quality of Service (QoS), and security. To mitigate malicious attacks, IoT systems typically employ attack detection to analyze network data, identify anomalous traffic, and intercept attack flows, thereby substantially reducing the success rate of network intrusions.

In recent years, numerous methods have demonstrated promising attack detection results for IoT. Nevertheless, identifying anomalous network traffic in this domain remains challenging. On one hand, IoT constitutes a vast, complex system characterized by heterogeneity, resource constraints, dynamicity, and privacy concerns. On the other hand, the extreme rarity of anomalous events in real-world scenarios leads to scarce attack traffic samples, resulting in long-tailed datasets for model training. These issues hinder the achievement of optimal anomaly detection performance.

To address these challenges and accurately identify malicious traffic targeting IoT devices, this paper proposes an IoT attack detection method based on the Genetic Algorithm (GA)[4] and Equalization Loss v2 (EQL v2)[5]. First, GA determines an optimal feature subset; second, EQL v2 resolves the scarcity of attack traffic samples; finally, a hybrid model combining Convolutional Neural Network (CNN) and Long Short-Term Memory Network (LSTM) extracts spatiotemporal information from network traffic.

In summary, the main contributions of this work to the IoT field are as follows:

(1) Class imbalance mitigation: EQL v2, a loss function with gradient-guided reweighting, is employed to address class imbalance in network traffic datasets, enhancing detection rates for rare attack classes.

(2) Feature selection optimization: A heuristic search algorithm identifies a minimally redundant feature subset that best approximates the original feature space.

(3) Spatiotemporal modeling: A novel CNN-LSTM hybrid detection model extracts temporal and spatial features to boost detection performance.

Naval University of Engineering, Wuhan 430033, China. ✉email: liutaotaoh@163.com

The remainder of this paper is organized as follows: Sect. “Related Work” discusses IoT attack detection techniques; Sect. “Methodology” briefly introduces the theoretical foundations of our method; Sect. "Datasets and preprocessing" describes benchmark datasets and preprocessing; Sect. “Experimental settings” details the experimental environment, evaluation metrics, and parameter settings; Sect. "Experimental results and analysis" presents results and comparative analyses; and Sect. “Conclusion” concludes the work and suggests future research directions.

Related work

This section first introduces attack detection models applied in IoT, then elaborates on recent research advancements in class imbalance proposed by domestic and international scholars.

IoT attacks

IoT attacks refer to network intrusions in which adversaries exploit any IoT device to access users’ sensitive data. Due to the lack of adequate security mechanisms in IoT systems, such attacks pose significant security risks.

The study in[6] provides a detailed overview of poisoning attacks commonly encountered in recommendation systems and categorizes them into four stages to identify the focal points of different types of poisoning attacks. In[7], to reveal the inherent vulnerabilities of federated recommendation systems, the authors conduct poisoning attacks by injecting a set of synthetically generated malicious users—without relying on any prior knowledge—in order to manipulate the exposure of target items. The work in[8] classifies IoT attacks based on multiple dimensions, including attack domains, threat types, execution methods, software surfaces, IoT protocols, device properties, adversary locations, and levels of information damage. It also proposes corresponding defense strategies to mitigate these threats.

Machine learning techniques

Machine learning has long played a significant role in IoT attack detection, with a wide variety of ML-based methods continuously emerging. For example, in[9], an improved Support Vector Machine (SVM) is employed for detection, and a binary Grey Wolf Optimization algorithm is used to select important features to reduce the false positive rate. In[10], a distributed ensemble intrusion detection system (IDS) based on fog computing for IoT is proposed, utilizing K-Nearest Neighbors (KNN), XGBoost, and Gaussian Naive Bayes as base classifiers, followed by classification using a Random Forest. In[11], a combination of Extra Trees and Random Forest is applied to IoT attack detection, achieving promising results. Reference[12] introduces an attack detection system integrating multiple SVMs, where each SVM is responsible for detecting a specific type of attack, enabling accurate detection and enhancing the security of IoT devices. However, with the increasing diversity of attack methods, shallow learning techniques represented by traditional machine learning are becoming insufficient to meet the evolving demands of network attack detection. Therefore, it is urgent to explore novel learning algorithms that can overcome the limitations of ML-based approaches.

Deep learning techniques

Given the significant advantages of deep learning in data mining, it has been widely applied in the field of attack detection. In[13], a knowledge distillation and deep metric learning-based IoT attack detection model is proposed, which not only improves anomaly detection accuracy but also significantly reduces model size and computational cost. In[14], a multi-frequency deep learning framework is constructed, leveraging the differences in occurrence frequencies of various data types. This framework combines high-frequency and low-frequency layers to build multi-frequency Transformer and LSTM modules for attack detection, achieving good results on IoT datasets. Reference[15] conducts multiple experiments using DNN and CNN models on feature-selected datasets to identify the most suitable model for anomaly detection, yielding fairly satisfactory outcomes. In[16], an adaptive optimization algorithm based on mutation and perception strategies is proposed, incorporating reinforcement learning into an SVM classifier, achieving detection rates of 99.71% and 99.61% on the NSL-KDD and CIC-IDS-2017 IoT benchmark datasets, respectively. However, the above methods overlook the temporal characteristics of network traffic. To address this, this paper constructs an IoT attack detection model based on Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. The CNN is used to extract spatial information from the network traffic of IoT devices, and LSTM is then introduced to capture temporal features. The proposed model effectively extracts deep features from network traffic and demonstrates strong robustness. Nevertheless, the issue of class imbalance remains unresolved, and further research is needed to address this challenge.

Class imbalance techniques

Class imbalance, a common issue in intrusion detection systems (IDS), has seriously affected system performance. Current solutions to this problem mainly fall into three categories: data-level methods, ensemble methods, and cost-sensitive methods.

Data-level methods aim to balance the dataset by modifying the data distribution and are currently the most widely used approach. For example[17], applied a K-Nearest Neighbors (KNN)-based under-sampling method to the majority class, which effectively balanced the data. However, data-level methods are susceptible to noise and may lead to information loss. In[18], a hybrid approach combining clustering-based SMOTE and K-Means under-sampling was proposed. This method avoids the high computational cost of SMOTE while mitigating the risk of losing critical information due to random under-sampling. Ensemble methods alleviate class imbalance by incorporating ensemble learning techniques. In[19], a dual ensemble model based on bagging and Gradient Boosting Decision Trees (GBDT) achieved promising results on three public datasets. In[20], parallel ensemble learning with DNN, XGBoost (XGB), and Gradient Boosting Machines (GBM) further improved anomaly

detection performance. However, ensemble methods often suffer from high computational cost and long processing times. Cost-sensitive methods address class imbalance by adjusting the loss function's weighting to reduce the false positive rate. For instance[21], employed focal loss to make the model focus more on minority classes, thereby preventing them from being overwhelmed by the majority classes. This led to improved detection performance across multiple datasets. Cost-sensitive approaches are simple, efficient, and do not alter the dataset.

Therefore, this paper introduces EQL v2 into the field of attack detection. By partitioning the classification problem into multiple independent tasks—each corresponding to one class—and reweighting their contributions based on a gradient-guided mechanism, EQL v2 balances the training process across classes. This loss function effectively addresses the class imbalance issue with relatively low computational cost.

Methodology

This section elaborates on the proposed theoretical framework, comprising four components: feature selection, spatiotemporal modeling, loss function design and integrated detection framework.

Feature selection

As a critical component in network attack detection, feature selection plays a fundamental role in enabling effective classification and recognition by subsequent models. To this end, this paper proposes an adaptive search method based on a genetic algorithm. Genetic algorithms (GA), first introduced by Holland in 1975[4], are computational models inspired by the process of natural evolution. They operate through continuous crossover and mutation within a population to preserve highly adaptive individuals, ultimately allowing the population to converge toward an optimal solution. The specific steps are as follows:

(1) Randomly initialize the population; (2) Evaluate whether the fitness of individuals in the population meets the optimization criteria. If it does, the search terminates and the optimal solution is output; otherwise, the process continues; (3) Select individuals based on fitness. Individuals with higher fitness are more likely to be selected; (4) Perform crossover operations on selected individuals, where parts of the parents' chromosomes are exchanged to produce new offspring; (5) Apply mutation operations to the new individuals to increase diversity within the population; (6) Repeat the above steps until a predefined maximum number of iterations is reached.

By following these steps, the algorithm can identify an optimal feature subset. This method is characterized by its simplicity, stability, and strong adaptability.

In the context of IoT anomaly detection, traffic datasets often contain redundant and high-dimensional features, many of which offer minimal contribution to classification and may introduce noise. Traditional feature selection methods, such as filter-based approaches, may fall into local optima and overlook complex nonlinear dependencies. In contrast, GA explores the global feature space and identifies feature subsets with strong discriminative power. This reduces computational overhead, enhances model generalization, and is particularly advantageous for resource-constrained IoT devices, where model compactness and efficiency are critical.

Spatiotemporal model

Our designed model concurrently captures temporal dynamics and spatial patterns in network traffic.

Convolutional neural network (CNN)

Proposed by LeCun et al. (1990)[22], CNN employs convolutional kernels to extract latent spatial features through sliding-window operations. Thus, we utilize CNN to process network traffic features represented as one-dimensional vectors. These features encode critical spatial behaviors, such as port activity, protocol usage, and service transitions, which are often indicative of attacks like port scanning and DoS floods. By employing multiple convolutional and pooling layers, the model effectively learns hierarchical representations of localized spatial anomalies in IoT traffic. The structure is shown in Fig. 1.

In CNNs, the mathematical formulation for extracting input features is defined as follows, with visual context provided by the architecture diagram:

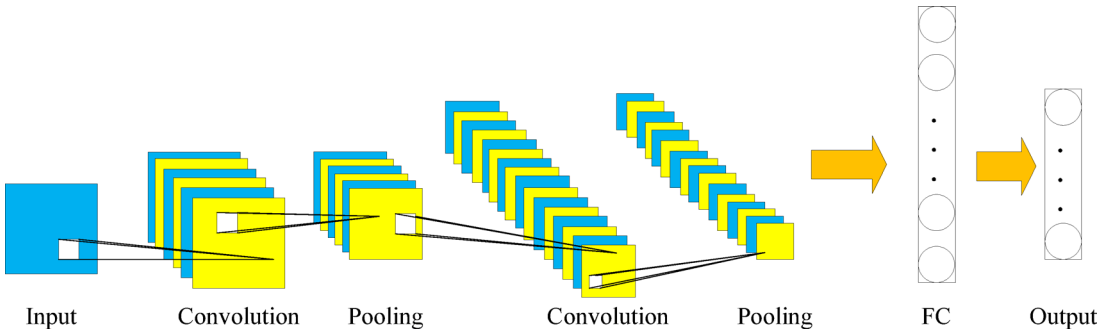

Fig. 1. Convolutional Neural Network (CNN) Architecture.

$$x_i^j = f(\sum_{i \in M_i} x_i^{(j-1)} * \omega_{il}^j + b_i^j) \tag{1}$$

In this context, $x_i^{(j-1)}$ represents the input feature map, x_i^j represents the output feature map, j denotes the current convolutional layer index, i and l indicate spatial positions, ω_{il}^j signifies the weight kernel, b_i^j is the bias term, and $f()$ is the activation function. Additionally, a pooling layer is typically appended after the convolutional layer. In this paper, average pooling is employed for dimensionality reduction, with its computational process defined in Eq. (2).

$$x_i = down(x_{i-1}) \tag{2}$$

Here, $x_i$represents the vector before pooling, $down()$denotes the pooling function, andx_{i-1} represents the vector after pooling.

Long Short-Term memory network (LSTM)

Long Short-Term Memory (LSTM) networks are a variant of Recurrent Neural Networks (RNNs)[23], designed to mitigate issues such as gradient vanishing. The architecture consists of an input layer, hidden layers, and an output layer.

In IoT scenarios, many attacks exhibit temporal continuity, such as slow brute-force attempts, lateral movement, or exfiltration over time. CNNs alone are insufficient to model such dependencies. Therefore, we introduce LSTM to capture time-sequential behaviors in traffic flow, enabling the detection of persistent or delayed attack patterns. The LSTM structure is shown in Fig. 2.

The key to LSTM's long-term memory capability lies in its gating mechanism and cell state, where f_t, i_t, and $o_t$represent the outputs of various gates in the network, x_t denotes the initial input to the network, h_t signifies the final output of the network, $\tilde{C}_t$indicates the temporary state of the cell, C_t represents the state of the cell at time step t, σ and $\tanh$ denote the activation functions.

EQL v2

To address class imbalance, a prevalent issue in IoT intrusion datasets, we employ EQL v2 as the loss function, which mitigates model neglect of minority samples during classification. EQL v2 effectively resolves class imbalance in NIDS through its gradient reweighting mechanism, balancing positive/negative gradient ratios. The positive and negative gradients of the loss $\mathcal{L}$ with respect to outputz_j are defined as:

$$\nabla_{z_j}^{pos}(\mathcal{L}) = \frac{1}{|L|} \sum_{i \in L} y_j^i (p_j^i - 1) \tag{3}$$

$$\nabla_{z_j}^{neg}(\mathcal{L}) = \frac{1}{|L|} \sum_{i \in L} (1 - y_j^i) p_j^i \tag{4}$$

In this context, p_j^irepresents the probability that the i-th instance belongs to class j, y denotes the true label, and L indicates the number of samples. To balance the positive and negative gradients, reweighting coefficients are defined as follows:

$$q_j^{(t)} = 1 + \alpha(1 - f(g_j^{(t)})), r_j^{(t)} = f(g_j^{(t)}) \tag{5}$$

Here, $q_j^{(t)}$ and$r_j^{(t)}$ represent the reweighting coefficients for positive and negative gradients respectively, $g_j^{(t)}$ denotes the accumulated ratio of positive to negative gradients after t iterations, $\alpha$is the balance coefficient, t indicates the iteration count, and $f()$is a constructed mapping function defined as follows:

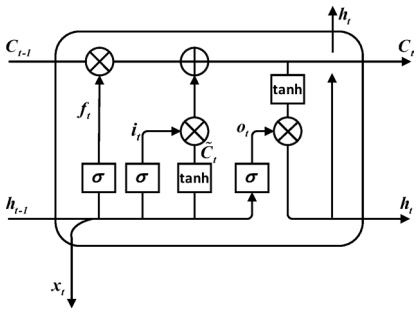

Fig. 2. Long Short-Term Memory (LSTM) Network Architecture.

$$f(x) = \frac{1}{1 + e^{-\gamma(x-\mu)}} \tag{6}$$

After obtaining the reweighting coefficients, the positive and negative gradients are updated as follows:

$$\nabla_{z_j}^{pos'}(\mathcal{L}^{(t)}) = q_j^{(t)} \nabla_{z_j}^{pos}(\mathcal{L}^{(t)}) \tag{7}$$

$$\nabla_{z_j}^{neg'}(\mathcal{L}^{(t)}) = r_j^{(t)} \nabla_{z_j}^{neg}(\mathcal{L}^{(t)}) \tag{8}$$

Additionally, the positive-negative gradient ratio for the next iteration $t+1$ should be updated as follows:

$$g_j^{(t+1)} = \frac{\sum_{t'=0}^{t} \left| \nabla_{z_j}^{pos'}(\mathcal{L}^{(t')}) \right|}{\sum_{t'=0}^{t} \left| \nabla_{z_j}^{neg'}(\mathcal{L}^{(t')}) \right|} \tag{9}$$

In summary, the aforementioned methodology effectively addresses the scarcity of anomalous traffic in IoT devices.

Proposed detection framework

This paper implements anomaly detection for IoT network traffic through Sect. 2.1–2.3, with the following concrete steps:

Step 1: Perform preprocessing operations including numerical encoding and normalization on the traffic dataset (details in Sect. 3.2).

Step 2: Execute feature selection using GA on the IoT traffic dataset.

Step 3: Construct a spatiotemporal model integrating CNN and LSTM, and configure parameters.

Step 4: Adopt EQL v2 as the loss function to balance positive/negative gradients and address class imbalance.

Step 5: Conduct preliminary training with K-fold cross-validation until model convergence.

Step 6: Input preprocessed test data into the trained model and analyze classification results.

Datasets and preprocessing

To validate the proposed method's efficacy for IoT attack detection, experiments are conducted on benchmark datasets: NSL-KDD[24] and CIC-IDS-2017[25].

Dataset description

The NSL-KDD dataset addresses the long-standing issues of the KDD CUP99 dataset by removing a large amount of redundant data, making it one of the most widely used benchmark datasets in the field of intrusion detection. It includes four files: KDDTrain+.txt, KDDTest+.txt, KDDTrain-21.txt, and KDDTest-21.txt. To better train the proposed method, this study selects KDDTrain+.txt and KDDTest+.txt as the training and testing sets, respectively. Compared with anomalous traffic, emerging IoT intrusion detection datasets contain a large number of benign samples. Therefore, the NSL-KDD dataset can be used to simulate network traffic in a realistic CPS environment, and evaluate our model by analyzing the detection of different anomalous traffic types. The details are shown in Table 1.

The CIC-IDS-2017 dataset was generated by sniffing real-time network packets and is one of the largest and most up-to-date intrusion detection datasets. Due to the diversity of IoT devices, they are highly vulnerable to various types of network attacks, including DDoS and SQL injection. The CIC-IDS-2017 dataset contains a large volume of DoS attacks, making it a suitable benchmark dataset for IoT security research. Due to hardware limitations, this study uses only the Wednesday portion of the CIC-IDS-2017 dataset for experiments. Additionally, since there are only 11 samples of the Heartbleed attack on Wednesday, which is too sparse, they are removed. The dataset still exhibits significant class imbalance, with details presented in Table 2.

Data preprocessing

To meet the input requirements of neural networks and improve the overall quality of the dataset, this study performs preprocessing on the data as follows:

(1) Numerical Encoding: Some features in the aforementioned datasets are non-numeric, such as protocol_type, flag, and service in the NSL-KDD dataset. Since the model cannot directly learn from categorical features,

Class	Train	Test
Normal	67,343	9711
Dos	45,927	7460
Probe	11,656	2421
R2L	995	2885
U2R	52	67
Sum	125,973	22,544

Table 1. Sample distribution of NSL-KDD Dataset.

Class	Train	Test
Benign	351,719	87,964
Dos Hulk	183,998	46,126
Dos GoldenEye	8307	1986
Dos Slowloris	4648	1148
Dos Slowhttptest	4444	1055
Sum	553,116	138,279

Table 2. Sample distribution of CIC-IDS-2017 Dataset.

CNN-LSTM	batchsize	128
	lr	0.001
	kernels	3
	epoch	5
	optim	Adam
	activation	ReLU
EQL v2	α	4
	γ	12
	μ	0.8

Table 3. Parameter Settings.

these need to be converted into numerical values. In this work, label encoding is applied to transform such features into numerical representations. The class labels are also converted into numerical form.

(2) Normalization: Certain feature dimensions in the dataset have a wide range of values, contributing unequally to the model's learning process. To reduce discrepancies between feature dimensions and ensure the accuracy of detection results, this study adopts Min-Max normalization for each feature column. This scales the values to the [0, 1] range and helps maintain data consistency and effectiveness.

Experimental settings

The computer used in this section is equipped with a 64-bit Windows 11 operating system, 16GB of RAM, an AMD Ryzen 7 6800 H CPU, and an NVIDIA GeForce RTX 3050Ti GPU.

Evaluation metrics

To comprehensively evaluate the model's performance and robustness, five metrics are employed:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{10}$$

$$Pre = \frac{TP}{TP + FP} \tag{11}$$

$$Rec = \frac{TP}{TP + FN} \tag{12}$$

$$FPR = \frac{FP}{FP + TN} \tag{13}$$

$$F1 = \frac{2 \times Pre \times Rec}{Pre + Rec} \tag{14}$$

Among them, TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively, that is, the number of correctly predicted attack samples, correctly predicted normal samples, incorrectly predicted attack samples, and incorrectly predicted normal samples.

Parameter settings

Extensive parameters exist throughout the feature selection to model classification pipeline, whose values critically impact final performance. Optimal parameter values were determined through comparative experimental analysis, with specific configurations detailed in Table 3.

Dataset	Acc	Pre	Rec	F1	FAR
NSL-KDD	99.21	99.22	99.21	99.22	4.18
CIC-IDS-2017	99.83	99.83	99.83	99.83	0.11

Table 4. Multi-class classification results (%).

(a)NSL-KDD (b)CIC-IDS-2017

Fig. 3. Confusion Matrix Visualization.

Dataset	Feature Selection	Acc	Pre	Rec	F1	FAR
NSL-KDD	Chi-square	97.16	97.04	97.16	96.84	5.98
	RF	98.98	99.00	98.98	98.97	2.80
	MI	97.45	97.44	97.45	97.41	3.17
	RFE	98.73	98.73	98.73	98.71	3.49
	GA (ours)	**99.21**	**99.22**	**99.21**	**99.22**	**4.18**
CIC-IDS-2017	Chi-square	95.05	95.54	95.05	95.08	5.12
	RF	99.82	99.82	99.82	99.82	0.15
	MI	99.73	99.73	99.73	99.73	0.19
	RFE	99.77	99.77	99.77	99.77	0.20
	GA (ours)	**99.83**	**99.83**	**99.83**	**99.83**	**0.11**

Table 5. Comparative analysis of feature selection methods (%). Bold Text Indicates Optimal Performance.

Experimental results and analysis

This section presents comprehensive experimental validation of the proposed method's effectiveness on IoT attack detection datasets NSL-KDD and CIC-IDS-2017. Through rigorous multi-phase testing and comparative studies, the performance advantages of our approach are systematically demonstrated.

Classification results experiment

For different attack patterns, corresponding defense strategies vary significantly. Merely identifying attack samples is insufficient to protect devices from compromise. Therefore, multi-class classification experiments are essential to categorize attack subtypes. The experimental results of the proposed method on both datasets and corresponding confusion matrices are presented in Table 4; Fig. 3, respectively.

As shown in Table 4, the proposed method achieves 99.21% accuracy and recall on NSL-KDD, with precision and F1-score at 99.22% and a False Alarm Rate (FAR) of 4.18%. For the CIC-IDS-2017 dataset, all five metrics reach 99.83% with an FAR of 0.11%. These results demonstrate the method's robust performance across both datasets, effectively identifying each attack category while maintaining resilience against false positives triggered by normal traffic fluctuations.

The confusion matrix is widely regarded as the most intuitive and fundamental tool for evaluating model performance. This paper visualizes inter-class distinctions through heatmap intensity, chromatic difference, and luminance. When the darkest cells align along the diagonal, this indicates correct predictions for the majority of samples. The confusion matrices for both datasets are illustrated in Fig. 3.

Feature selection method comparison

To validate the advantages of the proposed feature selection method, we compare it with four conventional methods: Chi-square test, Random Forest, Mutual Information, and Recursive Feature Elimination (RFE), as shown in Table 5.

As illustrated in Table 5, for the NSL-KDD dataset, the proposed method achieves superior performance on four out of five evaluation metrics (excluding FAR), with recall improving by 0.23–2.05%. For the CIC-IDS-2017 dataset, our method outperforms the others across all five metrics. This is primarily because the Chi-square test assumes feature independence, whereas the dataset contains complex nonlinear relationships—leading to potentially only locally relevant features being selected. Similarly, the mutual information and RFE methods also ignore correlations among features. Although Random Forest is an ensemble model, it can easily overfit on high-dimensional data, resulting in poor generalization of the selected features. In contrast, the method proposed in this paper uses a GA for feature selection on IoT attack detection datasets, which not only makes full use of the inherent statistical information in network traffic samples but also demonstrates both scientific rigor.

Comparison of class imbalance techniques

To address the issue of class imbalance in IoT environments, where abnormal events occur with low probability, the proposed class balancing method is compared with several representative existing techniques. The detailed results are presented in Table 6.

As shown in Table 6, the proposed class balancing method achieves strong performance on both datasets, effectively balancing the trade-off between precision and recall. For the NSL-KDD dataset, the proposed method achieves the best results across all five evaluation metrics, with maximum improvements of 20.95%, 6.88%, 31.80%, and 21.29% on the first four metrics, respectively. Similarly, on the CIC-IDS-2017 dataset, the proposed method also outperforms other class balancing approaches, demonstrating consistently superior performance across all five metrics and maintaining strong attack detection effectiveness.

The superiority of the proposed method over other class balancing approaches lies in the following aspects: ROS and SMOTE rely on simple data replication, which often leads to overfitting. Although methods like RUS + SMOTE and KMeans + SMOTE adopt various strategies to mitigate overfitting, the synthesized samples tend to be highly similar, thus the overfitting issue remains. ADASYN, which is based on K-nearest neighbors and assigns different weights to different classes for data generation, is vulnerable to outliers. These traditional sampling methods are essentially shallow learning techniques that lack the capability to deeply learn from the original data, making them ineffective in distinguishing noisy data. As a result, they tend to generate a significant amount of noisy samples, thereby degrading the model's performance. Although deep learning methods such as CVAE and CWGAN are able to learn the intrinsic distribution of sample data, they may generate meaningless samples and suffer from issues such as model collapse and gradient vanishing. In contrast, the proposed method employs the EQL v2 approach with a gradient-guided reweighting mechanism, which neither alters the overall distribution of the original dataset by synthesizing samples, nor suffers from exploding or vanishing gradients due to its ability to balance accumulated positive and negative gradients. As a result, it consistently achieves superior performance across various evaluation metrics.

Comparative analysis with existing methods

To further investigate the proposed method's advancement in IoT attack detection, we conduct benchmark comparisons against state-of-the-art models using different datasets. The comprehensive results are systematically presented in Table 7, demonstrating significant performance advantages across critical security metrics.

(1) KD-TCNN[13]: KD-TCNN extracts features using a triplet CNN-based knowledge distillation model to achieve classification.

(2) TLHA[29]: TLHA designs a three-layer hybrid model for intrusion detection.

(3) CNN-LSTM[30]: CNN-LSTM proposes an efficient hybrid IDS model that integrates a Black Widow Optimization algorithm with a convolutional neural network and long short-term memory network.

(4) MFNet[14]: MFNet identifies the multi-frequency nature of network traffic by constructing multi-frequency LSTM and multi-frequency Transformer modules.

(5) TACGAN[17]: TACGAN designs a Tabular Auxiliary Classifier Generative Adversarial Network model for attack sample oversampling, addressing the class imbalance problem.

Dataset	Imbalanced Algorithms	Acc	Pre	Rec	F1	FAR
NSL-KDD	ROS[[26]]	78.26	92.34	67.41	77.93	7.39
	SMOTE[[26]]	81.16	96.42	69.48	80.76	3.41
	ADASYN[[26]]	80.10	96.16	67.74	79.49	3.57
	CVAE[[26]]	85.97	97.39	77.43	86.27	2.74
	CWGAN[[27]]	90.34	96.74	85.92	91.01	3.83
	GA(ours)	**99.21**	**99.22**	**99.21**	**99.22**	**4.18**
CIC-IDS-2017	ROS[[18]]	99.76	99.81	99.76	99.77	-
	SMOTE[[18]]	99.76	**99.83**	99.76	99.77	-
	ADASYN[[18]]	99.76	**99.83**	99.76	99.77	-
	RUS + SMOTE[[28]]	99.72	99.81	99.72	99.75	-
	KMeans + SMOTE[[28]]	99.70	99.81	99.70	99.74	-
	GA(ours)	**99.83**	**99.83**	**99.83**	**99.83**	**0.11**

Table 6. Comparative analysis with other class balancing methods (%).

Dataset	Model	Acc	Pre	Rec	F1
NSL-KDD	KD-TCNN[[13]]	98.44	98.60	98.47	98.57
	TLHA[[29]]	97.94	97.90	97.90	93.40
	CNN-LSTM[[30]]	98.67	97.48	**100**	98.73
	MFNet[[14]]	-	76.78	75.01	73.18
	This paper	**99.21**	**99.22**	99.21	**99.22**
CIC-IDS-2017	TACGAN[[17]]	95.86	96.85	94.79	95.81
	SVM-GAC[[16]]	**99.93**	98.73	99.61	98.91
	KD-TCNN[[13]]	99.44	99.48	99.47	99.46
	BT-TPF [[31]]	99.60	99.60	99.60	99.60
	This paper	99.83	**99.83**	**99.83**	**99.83**

Table 7. Comparative analysis with existing methods (%).

Model	Acc	Pre	Rec	F1	Train time(s)	Inference time(s)
w/o G	98.78	98.79	98.78	98.78	1107.52	15.41
w/o C	96.49	96.24	96.49	96.04	**71.30**	**0.93**
w/o L	98.57	98.55	98.57	98.55	705.91	11.06
w/o E	98.52	98.58	98.52	98.53	798.34	11.35
Our	**99.21**	**99.22**	**99.21**	**99.22**	591.30	7.99

Table 8. Ablation study on NSL-KDD (%).

(6) SVM-GAC[16]: SVM-GAC introduces reinforcement learning into SVM to detect new types of network attacks, thereby improving detection performance.

(7) BT-TPF[31]: BT-TPF utilizes a Siamese network for feature dimensionality reduction of high-dimensional network traffic and employs knowledge distillation to reduce model complexity.

As can be seen from Table 7, compared with existing methods, the method proposed in this paper is generally optimal on three datasets. On the NSL-KDD dataset, the recall rate of our method is only slightly lower than that of the CNN model by 0.79%, but the precision rate is 1.74% higher. Furthermore, in terms of the comprehensive F1 score, our detection results are also superior to that model. For the CIC-IDS-2017 dataset, all models exhibit good detection performance. This is because the CIC-IDS-2017 dataset contains a large amount of data and is relatively simple. In this case, our method is only 0.1% lower in precision than SVM-GAC, but it is 3.97%, 0.39%, and 0.23% higher than TACGAN, KD-TCNN, and BT-TPF, respectively. Moreover, in terms of the F1 score, our method is 4.02%, 0.92%, 0.37%, and 0.23% higher than these four models, respectively.

The reason why the performance of other models is lower than that of the model in this paper lies in the fact that CNN-LSTM and PyDSC fail to effectively address the class imbalance issue in the dataset, resulting in detection performance inferior to other models. KD-TCNN and BT-TPF achieve lightweight anomaly detection models through knowledge distillation, but the reduction in parameters leads to a decline in performance, so these two models are inferior to the model in this paper in all four indicators. TLHA utilizes undersampling to synthesize samples, which results in low data quality and limited improvement in model performance. The SVM-GAC model utilizes an adaptive algorithm based on mutation and perception strategies to identify malicious features, achieving relatively significant results. However, the ability of SVM to extract deep information from samples is limited, so the performance of this model is slightly inferior. WGAN, MCGAN, FCWGAN, and TACGAN fail to solve the problem of model collapse, resulting in generated data that does not conform to the original distribution. In summary, the model in this paper can effectively identify specific attack methods and protect network devices from threats to a certain extent.

Ablation study

To verify the necessity of each component in the proposed method, we conduct ablation experiments on our model, and the proposed approach mainly consists of four components: GA, CNN, LSTM, and EQL v2, which are abbreviated as G, C, L, and E, respectively, for ease of presentation. The ablation experiments are conducted on the NSL-KDD dataset, and the detailed results are presented in Table 8.

As shown in Table 8, among all components, CNN has the greatest impact on the experimental results, leading to decreases of 2.72%, 2.98%, 2.72%, and 3.18% across the four evaluation metrics. This is primarily because CNN is capable of extracting high-quality spatial features, which provide a solid foundation for subsequent classifiers. The next most influential components are EQL v2 and LSTM, both resulting in approximately 0.7% performance degradation across all metrics. This can be attributed to EQL v2 effectively addressing class imbalance issues, while LSTM further improves the model by incorporating spatiotemporal traffic information. Finally, the use of GA not only eliminates redundant features, thereby reducing computational cost, but also enhances classification performance. In summary, the ablation study fully demonstrates the necessity of each component in the proposed method and highlights its overall effectiveness.

In addition, for resource-constrained IoT environments, timely attack detection and identification are equally critical. Therefore, we also analyzed the training time and inference time of the proposed method. As shown in Table 8, our approach achieves a relatively high ranking in both training and inference efficiency. Despite incorporating several additional components, the model is still able to detect network attacks in a timely manner. Although the variant without the CNN module exhibits slightly lower time consumption, this comes at the cost of reduced detection accuracy. Consequently, our method strikes a balance between detection performance and computational efficiency.

Conclusion

This paper proposes a deep learning-based attack detection method for the Internet of Things. By employing feature selection, we eliminate redundant features while enhancing inter-class discrimination and intra-class similarity within the optimized feature subset. Furthermore, we effectively address class imbalance in attack detection datasets through Equalization Loss v2 (EQL v2), simultaneously constructing a spatiotemporal model to extract temporal and spatial information from network traffic samples. The proposed method undergoes rigorous multi-class performance evaluation on two IoT attack detection benchmark datasets:NSL-KDD and CIC-IDS-2017. Experimental results demonstrate superior performance advantages over multiple comparative methods, confirming the approach's effectiveness and feasibility for real-world IoT security applications. This study is currently validated on only two traffic datasets, future research will aim to assess its adaptability to real-world heterogeneous IoT environments, and design the lightweight detection model to compatible with various mobile devices will constitute a critical focus in subsequent research.

Data availability

The NSL-KDD and CIC-IDS-2017 datasets used in this study are publicly available at https://www.unb.ca/cic/datasets/nsl.html and https://www.unb.ca/cic/datasets/ids-2017.html, respectively.

Received: 3 July 2025; Accepted: 4 August 2025
Published online: 06 August 2025

References

1. Gubbi, J. et al. Internet of things (IoT): A vision, architectural elements, and future directions[J]. *Future Generation Comput. Syst.* **29** (7), 1645–1660 (2013).
2. Ni, J. et al. Securing fog computing for internet of things applications: challenges and solutions[J]. *IEEE Commun. Surv. Tutorials.* **20** (1), 601–628 (2017).
3. Which? How a smart home could be at risk from hackers. Accessed: Nov. 1, 2022. [Online]. Available: https://www.which.co.uk/news/article/how-the-smart-home-could-be-at-risk-from-hackers-ak eR18s9eBHU.
4. Lambora, A., Gupta, K. & Chopra, K. Genetic algorithm-A literature review[C]//2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon). IEEE, : 380–384. (2019).
5. Tan, J. et al. Equalization loss v2: A new gradient balance approach for long-tailed object detection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. : 1685–1694. (2021).
6. Wang, Z., Yu, J., Gao, M., Yuan, W., Ye, G., Sadiq, S. and Yin, H. "Poisoning attacks and defenses in recommender systems: A survey," CoRR, vol. abs/2406.01022 (2024).
7. Yuan, W. et al. Manipulating federated recommender systems: Poisoning with synthetic users and its countermeasures[C]// Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. : 1690–1699. (2023).
8. Sasi, T. et al. A comprehensive survey on IoT attacks: taxonomy, detection mechanisms and challenges[J]. *J. Inform. Intell.* **2** (6), 455–513 (2024).
9. Safaldin, M., Otair, M. & Abualigah, L. Improved binary Gray Wolf optimizer and SVM for intrusion detection system in wireless sensor networks[J]. *J. Ambient Intell. Humaniz. Comput.* **12**, 1559–1576 (2021).
10. Kumar, P., Gupta, G. P. & Tripathi, R. A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks[J]. *J. Ambient Intell. Humaniz. Comput.* **12** (10), 9555–9572 (2021).
11. De Souza, C. A., Westphall, C. B. & Machado, R. B. Two-step ensemble approach for intrusion detection and identification in IoT and fog computing environments[J]. *Comput. Electr. Eng.* **98**, 107694 (2022).
12. Vijayanand, R., Devaraj, D. & Kannapiran, B. Support vector machine based intrusion detection system with reduced input features for advanced metering infrastructure of smart grid[C]//2017 4th International conference on advanced computing and communication systems (ICACCS). IEEE, : 1–7. (2017).
13. Wang, Z. et al. A lightweight approach for network intrusion detection in industrial cyber-physical systems based on knowledge distillation and deep metric learning[J]. *Expert Syst. Appl.* **206**, 117671 (2022).
14. Ding, Z. et al. MF-Net: Multi-frequency intrusion detection network for internet traffic data[J]. *Pattern Recogn.* **146**, 109999 (2024).
15. Al-Turaiki, I. & Altwaijry, N. A convolutional neural network for improved anomaly-based network intrusion detection[J]. *Big Data*. **9** (3), 233–252 (2021).
16. Shukla, A. K. Detection of anomaly intrusion utilizing self-adaptive grasshopper optimization algorithm[J]. *Neural Comput. Appl.* **33** (13), 7541–7561 (2021).
17. Ding, H. et al. Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection[J]. *Future Generation Comput. Syst.* **131**, 240–254 (2022).
18. Song, J. et al. CSK-CNN: network intrusion detection model based on Two-Layer Convolution neural network for handling imbalanced Dataset[J]. *Information* **14** (2), 130 (2023).
19. Louk, M. H. L. & Tama, B. A. Dual-IDS: A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system[J]. *Expert Syst. Appl.* **213**, 119030 (2023).
20. Vo, H. V., Du, H. P. & Nguyen, H. N. AI-powered intrusion detection in large-scale traffic networks based on flow sensing strategy and parallel deep analysis[J]. *J. Netw. Comput. Appl.* **220**, 103735 (2023).
21. Khan, N. M. et al. Analysis on improving the performance of machine learning models using feature selection technique[C]// Intelligent Systems Design and Applications: 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018) held in Vellore, India, December 6–8, Volume 2. Springer International Publishing, 2020: 69–77. (2018).
22. LeCun, Y. et al. Handwritten digit recognition with a back-propagation network[J]. *Adv. Neural. Inf. Process. Syst.*, 2. (1989).

23. Hochreiter S, Schmidhuber J. Long short-term memory[J]. *Neural computation*, **9**(8), 1735–1780 (1997).
24. Dhanabal, L. & Shantharajah, S. P. A study on NSL-KDD dataset for intrusion detection system based on classification algorithms[J]. *Int. J. Adv. Res. Comput. Communication Eng.* **4** (6), 446–452 (2015).
25. Sharafaldin, I., Lashkari, A. H. & Ghorbani, A. A. Toward generating a new intrusion detection dataset and intrusion traffic characterization[J]. *ICISSp* **1**, 108–116 (2018).
26. Yang, Y. et al. Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network[J]. *Sensors* **19** (11), 2528 (2019).
27. Zhang, G. et al. Network intrusion detection based on conditional Wasserstein generative adversarial network and cost-sensitive stacked autoencoder[J]. *IEEE Access.* **8**, 190431–190447 (2020).
28. Zhang, H. et al. An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset[J]. *Comput. Netw.* **177**, 107315 (2020).
29. Harini, R. et al. An effective technique for detecting minority attacks in NIDS using deep learning and sampling approach[J]. *Alexandria Eng. J.* **78**, 469–482 (2023).
30. Kanna, P. R. & Santhi, P. Hybrid intrusion detection using mapreduce based black widow optimized convolutional long short-term memory neural networks[J]. *Expert Syst. Appl.* **194**, 116545 (2022).
31. Wang, Z. et al. A lightweight IoT intrusion detection model based on improved BERT-of-Theseus[J]. *Expert Syst. Appl.* **238**, 122045 (2024).

Author contributions

Yu wrote the main manuscript textFu ConceptualizationLiu Software and MethodlogyWang Review & editingAn Review & editing.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to T.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.