A Lightweight Multi-Classification Intrusion Detection Model for Edge IoT Networks

Wei Gao , Mingyue Wang , Yadong Pei , Fangwei Li , Chaonan Wang *

Posted Date: 2 February 2026

doi: 10.20944/preprints202601.2418.v1

Keywords: IoT security; intrusion detection; feature reduction; temporal convolution networks; mutual information; pearson correlation coefficient

 doi:10.20944/preprints202601.2418.v1

Article

A Lightweight Multi-Classification Intrusion Detection Model for Edge IoT Networks

Wei Gao [1,2], Mingyue Wang [1,2], Yadong Pei [1,2], Fangwei Li [3] and Chaonan Wang [1,2,*]

[1] Chongqing College of Mobile Communication, Chongqing 401420, China
[2] Chongqing Key Laboratory of Public Big Data Security Technology, Chongqing 401420, China
[3] Chongqing University of Posts and Telecommunications
* Correspondence: wangchaonan926@163.com

Abstract

Intrusion detection aims to effectively detect abnormal attacks in the Internet of Things (IoT) network, which is crucial for cybersecurity. However, traditional intrusion detection methods are difficult to effectively extract data features from traffic data, and most existing models are too complex to be deployed on edge servers. Addressing this need, this paper proposes a hybrid feature selection method and a lightweight deep learning intrusion detection model. Firstly, the data features space is reduced by using variance filtering, mutual information, and the Pearson Correlation Coefficient, thereby reducing the computational cost of subsequent model training. Then, an intrusion detection model based on Temporal Convolutional Network (TCN) is constructed. This model utilizes dilated causal convolutions to effectively capture long-term temporal dependencies in network traffic. Simultaneously, use the residual connections to mitigate the vanishing gradient problem, making the model easier to train and converge. Finally, experiments are conducted on the newly released Edge-IIoTset dataset. The results show that the proposed feature selection algorithm maintains good detection performance despite a significant reduction in feature dimensionality. Furthermore, compared with other models, the proposed TCN-based approach achieves higher classification accuracy with lower computational overhead, demonstrating its suitability for deployment in resource-constrained edge computing environments.

Keywords: IoT security; intrusion detection; feature reduction; temporal convolution networks; mutual information; pearson correlation coefficient

1. Introduction

With the continuous advancement of information technology, the Internet of Things (IoT) has gradually integrated into all aspects of society and has shown extensive application value in many fields, such as industrial IoT [1], healthcare [2], smart cities [3], and smart homes [4]. However, due to the characteristics of IoT devices, including simplified hardware design, limited computing and storage resources, diverse network protocols, and decentralized geographical deployment, IoT devices are extremely vulnerable to network attacks in practical applications, causing serious network security incidents such as network system paralysis and user privacy leaks [5,6]. Therefore, how to effectively ensure the security and stable operation of IoT systems has become an important issue that urgently needs to be addressed.

In order to deal with the threat of network attacks, network intrusion detection system (NIDS) has emerged as an important part of the proactive defense system. NIDS can identify potential attack behaviors, and abnormal access by real-time monitoring and analysis of system logs, user behavior and network traffic. Research on NIDS began globally in the 1980s and has now become an indispensable part of network security [7]. However, with the continuous development of digital technology and the increasing diversification of network attack methods, the data characteristics of

 doi:10.20944/preprints202601.2418.v1

network traffic have become more and more complex, and traditional NIDS can no longer meet the needs of IoT users to accurately identify anomalies and unknown attacks [8].

With the development of artificial intelligence, the NIDS field has begun to use deep learning to extract traffic features, thereby automatically identifying abnormal traffic in complex data traffic. Commonly used deep learning algorithms mainly include Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory Networks (LSTM) [9]. However, these methods generally suffer from complex network structures and large parameter scales, thus requiring high hardware resources for the deployment environment. Meanwhile, traditional NIDS are mostly deployed in the cloud or on central servers. Although they possess strong computing and storage capabilities, they suffer from high latency and high bandwidth consumption in IoT scenarios. To address this, researchers have proposed deploying intrusion detection systems at the edge, enabling traffic analysis and detection to be completed locally. This effectively reduces communication latency, alleviates network transmission pressure, and improves the response speed to local attacks [10].

The introduction of edge-based intrusion detection provides a new assurance for the real-time performance and security of IoT networks. However, it also brings several challenges that urgently need to be addressed:

1. IoT network traffic data usually exhibits high dimensionality and strong feature correlation. If such data are directly fed into the detection model, redundant information will significantly increase computational burden and make the model prone to overfitting, which in turn degrades detection accuracy.
2. Most deep learning–based intrusion detection models require substantial computational and storage resources. Given the resource-constrained nature of edge devices, models with large parameter sizes and high computational costs are difficult to deploy and run efficiently in practical edge environments.

To address the aforementioned issues, this paper aims to research a lightweight intrusion detection method for the IoT at the edge. The main contributions are as follows:

1. A hybrid feature extraction strategy: Aiming at the issues of high data dimensionality and feature redundancy in IoT network traffic, a hybrid feature selection method integrating variance filtering, mutual information, and correlation analysis is proposed. Firstly, variance filtering is used to quickly remove the information features with small variation amplitude and limited discrimination ability, thereby reducing the computational overhead of subsequent feature selection. Then, a feature selection objective function is constructed by combining mutual information and Pearson Correlation Coefficient to select the feature subset with high correlation to the category label and low feature redundancy.
2. A lightweight intrusion detection model: A TCN-based intrusion detection model was developed, which effectively captures the long-term temporal dependency features of network traffic by leveraging its hierarchical causal convolution and dilated convolution structure. Meanwhile, residual connections are utilized to alleviate the vanishing gradient problem, making the model easy to train and converge. Compared with deep learning models, this method has a smaller parameter scale, faster convergence, and the ability to comprehensively extract sequence features, achieving good detection performance. It can be deployed on resource-constrained edge devices.
3. Validation on the latest dataset: Experimental results on the Edge-IIoTset dataset show that the proposed hybrid feature selection method can significantly reduce feature dimensionality while still maintaining good detection performance. At the same time, compared with other models, TCN has a advantage in classification and detection performance, which verifies the effectiveness of the model.

 doi:10.20944/preprints202601.2418.v1

2. Related Works

2.1. Feature Selection Methods for IoT Intrusion Detection

Given that IoT traffic data contains a large amount of redundant and irrelevant features, directly feeding the data into a model for training not only increases the model's computational complexity but also places high demands on the hardware of the deployment environment. To address this issue, many researchers have adopted feature selection methods to achieve data dimensionality reduction.

Traditional feature selection methods are primarily categorised into: filter methods, wrapper methods, and embedded methods [11]. During the feature selection process, the chosen feature subset should has strong classification ability while minimising redun-dancy between features. However, individual feature selection methods may perform well under specific conditions, their evaluation methods are too simplistic. Omuya E O et al. [12] proposed a feature dimensionality reduction method based on principal component analysis. This method requires constructing the covariance matrix of the data and performing eigenvalue decomposition, which has a large computational cost and is difficult to meet the real-time processing requirements of large-scale, high-dimensional IoT network traffic data. To reduce computational complexity, some studies adopt feature selection methods based on statistical properties. Alhassan et al. [13] introduced a novel intrusion detection method based on correlation feature subset selection by calculating the correlation between features and labels. Its detection accuracy is better than that of traditional intrusion detection systems that do not perform feature selection. However, this method only considers the linear correlation between features and categories and fails to consider the nonlinear correlations that exist in network data. Devaraju et al. investigated a feature selection algorithm based on information entropy [14]. This method measures the nonlinear relationship between features and labels by calculating the mutual information value between them. However, in scenarios with small samples or imbalanced class distributions, the stability of mutual information estimation is poor, and the feature selection performance is easily affected.

The mRMR method, by jointly considering the maximum correlation and minimum redundancy, show that the inspection performance of the selected feature subset is significantly better than the methods only use correlation or mutual information [15]. Nevertheless, this method only evaluates the correlation between features in the subset of candidate features, without considering the correlation between features and categories. Furthermore, since this method requires calculating the mutual information between features and category labels, it incurs high computational complexity when the sample size is large.

2.2. IoT Intrusion Detection Method Based on Deep Learning

In recent years, with the development of artificial intelligence, NIDS based on deep learning has gradually become a research hotspot. Deep learning by constructs multi-layer neural networks to realize the layer-by-layer feature representation. This approach can automatically extract key features from the original network traffic and capture complex temporal and spatial patterns, thereby achieving effective differentiation between normal and abnormal traffic [16,17].

Ullah et al. in reference [18] introduced a RNN-based method for intrusion detection in IoT networks. A multi-layer RNN architecture is designed to exploit the capability of RNNs in modeling temporal sequences, enabling effective capture of time-dependent patterns and anomalous behaviors in network traffic. Reference [19] investigates six deep learning–based models for network attack detection, including multilayer perceptron (MLP), one-dimensional convolutional neural network (1D-CNN), long short-term memory (LSTM), gated recurrent unit (GRU), recurrent neural network (RNN), and a hybrid CNN–GRU model. The six models are systematically evaluated to compare their performance in classifying network traffic into normal or attack. In [20], an intrusion detection scheme is proposed for IoT devices. The authors encode sensor data through convolution operations to capture patterns within time series, integrating two classical CNN architectures: ResNet and EfficientNet, and evaluate their detection performance. Although these models can capture time-

 doi:10.20944/preprints202601.2418.v1

series dependencies in network traffic, they generally suffer from problems such as high detection latency, high computational complexity, and insufficient parallel processing capabilities.

To overcome the limitations of traditional recurrent neural networks in training efficiency and long-distance dependency capture, Temporal Convolutional Network (TCN) has been proposed as a new sequence modeling architecture. Compared with traditional deep learning models, TCN uses causal convolution and dilated convolution, combined with residual connections, which enables the network to efficiently capture long sequence dependencies, achieve gradient stable training, and support parallel computing. Inspired by the successful application of temporal convolutional networks in other fields, Lopes et al. in reference [21] designed and implemented four network intrusion detection models based on temporal convolution and conducted a systematic study on their classification performance. Ref. [22] proposed a network intrusion detection method based on temporal convolutional network (TCN). Experimental results show that TCN can effectively capture the temporal dependencies of complex network traffic and exhibit efficient and accurate detection performance. Ref. [23] innovatively combined causal and non-causal TCN to construct a hybrid intrusion detection system for the real-time and global prediction needs of IoT. The results show that hybrid TCN improves accuracy by 1.5% and recall by 4% on NSL-KDD, with only a marginal increase of 0.1 ms in detection time. However, most existing studies focus on limited attack types, and the used datasets do not comprehensively reflect the variety of IoT attacks.

In summary, although existing feature extraction techniques and intrusion detection models have achieved some progress, they still suffer from high computational overhead, suboptimal detection efficiency, and the incomplete coverage of attack types in commonly used datasets. To address these issues, this paper first introduces a hybrid feature selection algorithm to select optimal feature subsets. Subsequently, a lightweight intrusion detection model based on TCN is constructed, and its effectiveness is validated on the latest Edge-IIoTset dataset. This provides a feasible solution for deploying IoT intrusion detection systems on edge servers.

3. Proposed Methodologies

This section presents the overall methodological framework of the proposed approach. First, it describes the dataset and data preprocessing. After that,it briefly discusses the hybrid feature selection process. Finally, the training process of the TCN model is introduced. A block diagram of the proposed framework is presented in Figure 1

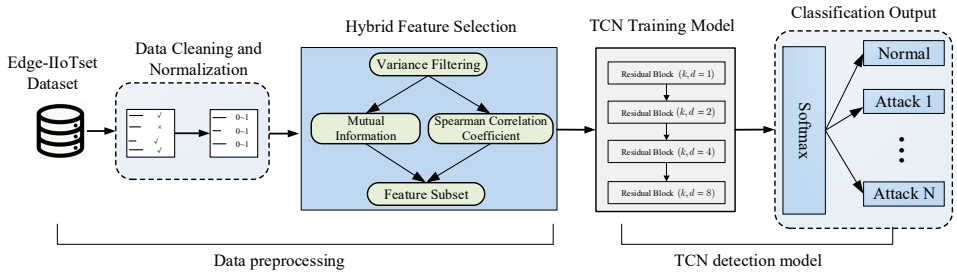

Figure 1. Overall process of the proposed model.

3.1. Dataset Description

In this study, we adopt the Edge-IIoTset dataset [24], which is widely used for security research in both IoT and Industrial IoT (IIoT) scenarios. The dataset was collected by Ferrag et al. (2022) from a realistic IIoT testbed environment that involves more than ten types of IoT devices. For experimental evaluation, we select the DNN-EdgeIIoT-dataset.csv file from the Edge-IIoTset dataset as the benchmark dataset. This subset provides a large-scale and diverse collection of traffic samples, making it suitable for training and evaluating deep learning–based intrusion detection models. The

 doi:10.20944/preprints202601.2418.v1

dataset contains a total of 2,219,201 network flow records, each described by 61 features, covering 14 different attack categories as well as normal traffic.

3.2. Dataset Preprocessing

To ensure data quality and enhance model training effectiveness, the raw dataset was preprocessed before model construction. As described in Ref. [25], we preprocess the dataset in the following manner. First, duplicate rows were removed, resulting in the removal of 815 redundant entries. Second, we removed unnecessary feature columns that did not significantly contribute to output predictions. These feature include (1)frame.time, (2)ip.dst_host, (3)http.file_data, (4)ip.src_host, (5)arp.src.proto_ipv4, (6)arp.dst.proto_ipv4, (7)http.request.full_uri, (8)http.request.uri.query, (9)icmp.transmit_timestamp, (10)tcp.payload, (11)tcp.options, (12)tcp.srcport, (13)tcp.dstport, (14)udp.port, and (15)mqtt.msg. After removal of these columns, the dataset contained 1,909,671 records with 47 remaining features. Third, in the new dataset, excluding Attack_type, seven columns contains object data type. These columns include (1) http.request.method, (2)http.referer, (3)http.request.version, (4)dns.qry.name.len, (5)mqtt.conack.flags, (6)mqtt.protoname and (7)mqtt.topic. To facilitate model training, these columns were transformed into binary 0–1 representations through dummy encoding. After encoding, the dataset contained 1,909,671 samples with 97 feature dimensions. Finally, since the original network traffic features exhibited varying value ranges, numerical standardization was performed to ensure that all features shared the same scale. The min–max normalization method was applied, which as described by the following formula:

$$f' = \frac{f - f_{\min}}{f_{\max} - f_{\min}} \tag{1}$$

where f' is the value of f after data normalization.

3.3. Hybrid Feature Selection Algorithm

Figure 2 illustrates the entire feature selection method, which is mainly divided into two modules: preliminary selection based on variance filtering and final selection based on statistical methods. The entire process starts with the original feature set $F = \{f_1, \ldots f_p\}$, and after variance filtering and statistical methods, it finally outputs a feature subset $F = \{f_1, \ldots f_K\}$. This section will describe in detail the principles of these two feature selection modules.

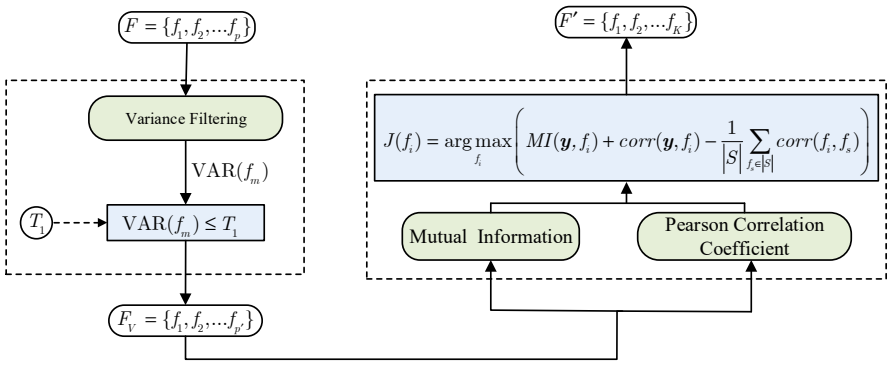

Figure 2. Feature selection process.

 doi:10.20944/preprints202601.2418.v1

3.3.1. Variance-Based Feature Selection

Variance filtering is a statistical feature selection method. Its core idea is to filter and reduce the dimensionality of features by calculating the variance of each feature in the sample space. When the variance is small, it indicates that the value of the feature does not vary much among different samples, the distribution is relatively concentrated, and it lacks sufficient discriminative power; conversely, it indicates that the feature shows significant differences among different samples and has stronger information expression potential. Let f_j be the j-th feature, and its variance can be defined as:

$$Var(f_j) = \frac{1}{N}\sum_{i=1}^{N}(f_{ij} - \mu_j)^2 \tag{2}$$

where N denote the number of samples, μ_j denote the mean of the j th feature. By using variance filtering to perform preliminary feature selection, initial dimensionality reduction of network data stream is achieved, which can effectively reduce the complexity of subsequent feature selection.

3.3.2. Feature Selection Based on Mutual Information and Correlation

The variance filtering relies solely on the statistical distribution characteristics of the network data , without considering the relationship between features and labels. To address this issue, mutual information and Pearson Correlation Coefficients are used to measure the relationship between them. Mutual information, a fundamental concept in information theory, has been employed to quantify the dependency between two variables. It measures the information that one variable conveys about the other. When knowledge of one variable enables accurate prediction of the other, the mutual information between them is considered as high [26]. Assume that $\boldsymbol{y}$ denotes the corresponding class label in the dataset. The mutual information between the class label and a feature can be expressed as:

$$MI(\boldsymbol{y}, f_i) = H(\boldsymbol{y}) + H(f_i) - H(\boldsymbol{y}, f_i) \tag{3}$$

where $H(\boldsymbol{y})$ and $H(f_i)$ denote the information entropies of the $\boldsymbol{y}$ and f_i respectively, and $H(\boldsymbol{y}, f_i)$ represents their joint entropy. As shown in Formula 3, a higher mutual information value suggests that a feature preserves more label-related information and thus contribute more effectively to classification performance.

Mutual information is commonly used to quantify the dependency between features and labels. However, this approach does not constrain redundancy among features. To alleviate this issue, the Pearson Correlation Coefficient is introduced to characterize the relationship between them. This method is called mRMR [27]. The Pearson Correlation Coefficient is defined as follows:

$$corr(\boldsymbol{y}, f_i) = \frac{\sum_{t=1}^{u}(\boldsymbol{y}_t - \overline{\boldsymbol{y}})(f_{i_t} - \overline{f_i})}{\sqrt{\sum_{t=1}^{u}(\boldsymbol{y}_t - \overline{\boldsymbol{y}})^2(f_{i_t} - \overline{f_i})^2}} \tag{4}$$

Moreover, correlations among features and labels can provide additional information to guide the selection of the features Consequently, during this process, the Pearson Correlation Coefficient is also employed to evaluate the features and labels. Based on the above considerations, the overall objective function of the proposed feature selection strategy is formulated as:

 doi:10.20944/preprints202601.2418.v1

$$J(f_i) = \arg\max_{f_i} \left(MI(\boldsymbol{y}, f_i) + corr(\boldsymbol{y}, f_i) - \frac{1}{|S|} \sum_{f_s \in |S|} \frac{corr(f_i, f_s)}{corr(\boldsymbol{y}, f_i)} \right) \quad (5)$$

where S denotes the already selected feature subset, and f_i represents a candidate feature.

3.3.3. Implementation of the Hybrid Method

Based on the feature selection method described above, a subset of features is selected from the preprocessed dataset. This subset ensures that the features are highly informative and have low correlation. The detailed algorithm is as follows:

Hybrid Feature Selection Algorithm

Input: Feature set $F = \{f_1, \ldots f_p\}$,class labels $\boldsymbol{y}$, Number of feature subsets: K

Output: The feature subset F'

Step1. Initialization: set $F' = \varnothing$

Step2. The variance filtering method is used to initially screen the original dataset, resulting in a new dataset $F_V = \{f_1, f_2, \ldots f_{p'}\}$

Step3. Calculate $MI(\boldsymbol{y}, f_i)$ for each feature in dataset F_V

Step4. Select the feature f_i with the highest mutual information

Step5. Set $F \leftarrow F \setminus f_i$, $F' \leftarrow F' \bigcup f_i$,

Step6. While $|F'| < K$ do

Calculate $J(f_i)$ in (5) to find $f_i, F \leftarrow F \setminus f_i, F' \leftarrow F' \bigcup f_i$

End For

Step7. Return F'

3.4. Intrusion Detection Framework Based on TCN

To effectively solve the time series modeling problem, TCN has shown good performance in time series data modeling tasks by introducing one-dimensional convolution, causal convolution, dilated convolution and residual structure. Based on the time series characteristics of network traffic in IoT systems, this paper constructs an intrusion detection model based on TCN, and its model diagram is shown in Figure 3. It contains following three parts.

 doi:10.20944/preprints202601.2418.v1

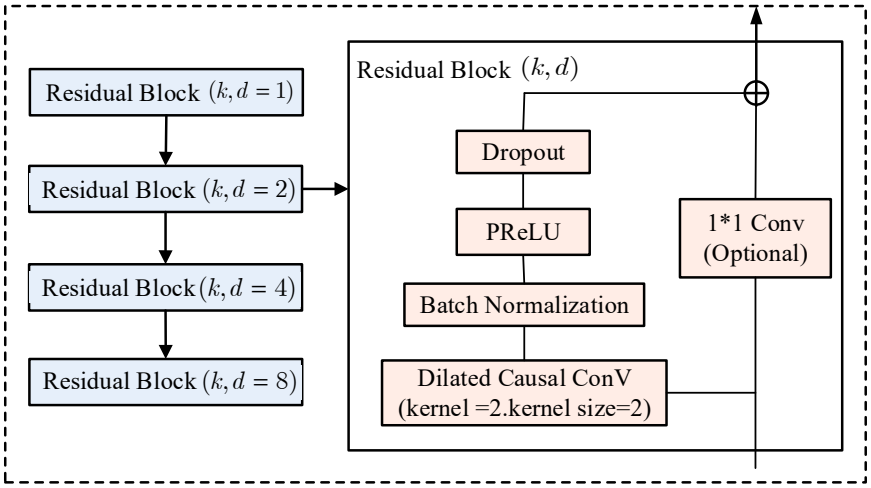

Figure 3. TCN model diagram.

3.4.1. Causal Convolution

Causal convolution restricts the sliding direction of the convolution kernel in the time dimension, ensuring that the output at each time step depends only on the current time step and the previous inputs. Therefore, using causal convolution, the intrusion detection output at time t is only convolved with the time series at time t or earlier, without losing historical data, and can preserve data traffic information over a longer period of time.

3.4.2. Dilated Convolution

Dilated convolutional layers allow for parallel computation of multiple time steps within a convolutional filter, while introducing an interval to skip certain elements during computation. Therefore, as the filter slides, it can compute the output at multiple locations simultaneously. This effectively enhances the model's predictive efficiency. Combining causal convolution with dilated convolution, it is defined as:

$$G(s) = (x \cdot g_d)(s) = \sum_{i=0}^{w-1} g(i) \cdot x_{s-d \cdot i} \tag{6}$$

where g_d denotes the convolution operation with a dilation factor d, w represents the kernel size, and $s - d \cdot i$ denotes the direction toward the past. The dilated convolution controls the receptive field by adjusting the dilation factor. When $d = 1$ it degenerates into a standard convolution. The figure 4 illustrates a dilated causal convolution with a dilation factor $d = 1, 2, 4$ and a kernel size of 2. By increasing the dilation factor, the receptive field can be expanded, which allows the model to capture longer temporal dependencies while reducing network depth without increasing the number of parameters.

 doi:10.20944/preprints202601.2418.v1

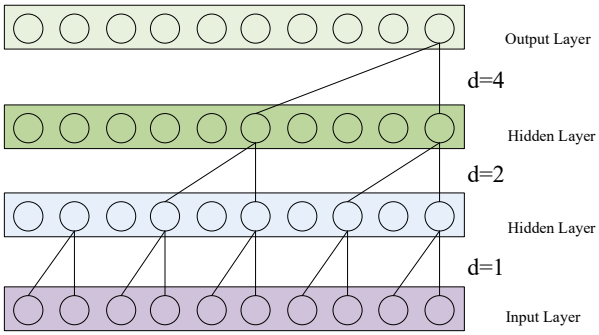

Figure 4. Illustration of the dilated causal convolution.

3.4.3. Residual Connection

The residual block simplifies the network training process by introducing residual connections. It typically consists of a series of convolutional layers and shortcut connections, enabling the model to capture long-range dependencies with fewer layers. This design facilitates easier training and faster convergence, mitigates the vanishing gradient problem in deep learning, and enhances both the training stability and generalization performance of the network. The residual block is composed of two branches that operate on the input and output respectively, which can be expressed as follows:

$$o = \delta(x + \varphi(x)) \tag{7}$$

where $\delta(\cdot)$ denotes the activation function, and $\varphi(x)$ represents the transformation composed of four components: dilated causal convolution, weight normalization, activation function, and dropout.

4. Experiments and Performance Evaluation

This section presents the experimental methodology and performance evaluation of the proposed approach. First, the implementation platform, performance evaluation metrics, and hyperparameters used for model training are described. Then, the experimental results of hybrid feature selection and TCN-based classification are discussed in detail.

4.1. Experimental Environment

The experiments were conducted on a workstation equipped with an AMD Ryzen 9 5900X 12-Core Processor processor and 32 of RAM. The software environment includes Windows 11, Python 3.9.13, and key libraries such as Tensorflow and Keras versions are 2.15.0.

4.2. Hyperparameters for Model's Training

In the training process of deep learning models, hyperparameter settings play a decisive role in the model's convergence speed, performance, and generalization ability. To obtain the best detection results on the dataset used, the experimental parameter settings in this paper are shown in Table 1.

Table 1. Model Hyperparameter Configuration.

Parameters	Value
Learning rate	0.001
Optimizer	Adam
Activation function	PReLU
Epochs	50
Batch_size	128
Dilation factor	8
Loss function	Categorical_crossentropy

 doi:10.20944/preprints202601.2418.v1

4.3. Performance Evaluation Metric

To evaluate the effectiveness of the created model, this paper uses Accuracy, Precision, Recall, and F1-score as evaluation metrics for the model training results. The specific calculation formulas are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + \mathrm{Re}\,call} \tag{11}$$

where TP denotes the number of correctly classified normal instances, TN represents the number of correctly classified attack instances, FP indicates the number of misclassified attack instances, and FN refers to the number of misclassified normal instances.

4.4. Hybrid Feature Selection Algorithm Experiment

In the first step of the hybrid feature selection algorithm, we use the median of the feature variances as the threshold value T_1. The calculation method is as follows. The variance of each feature in the dataset is calculated, after which these values are sorted in ascending order to identify the median as the threshold T_1. Features with variances below this value are removed. After this processing, the number of the features is reduced from 94 to 47. This step effectively reduces the feature dimensionality and alleviate the computational burden for subsequent feature selection.

To determine the final number of feature subset K, we selected 6, 12, 16, 18, 20, 22 and all features for experiment. TCN was used as the classifier. Experimental results in table2 show that as the number of features increases, the overall model performance improves, but the computational complexity also increases. Considering both detection performance and computational complexity, we ultimately chose 18 as the feature subset number.

Table 2. Detection results of different feature dimensions (values are in %).

Feature selection methods	Accuracy /%	Precision /%	Recall /%	F1 /%	Model parameters /KB
6	0.79	0.75	0.79	0.76	14.25
12	89.75	90.65	90.65	89.27	15
16	92.35	92.37	92.35	91.52	15.5
18	93.55	92.95	93.55	92.67	15.75
20	93.66	93.16	93.66	92.78	16
22	93.79	93.33	93.79	93.13	16.25
All features	94.24	93.77	94.24	93.71	25.37

 doi:10.20944/preprints202601.2418.v1

Figures 5 and 6 show the feature correlation heatmaps before and after mutual information and correlation feature selection. Figure 5 shows that after variance filtering, there are many correlated features among the network data stream features. Figure 6 demonstrates that after mutual information and correlation feature selection, the algorithm removes overly correlated features.

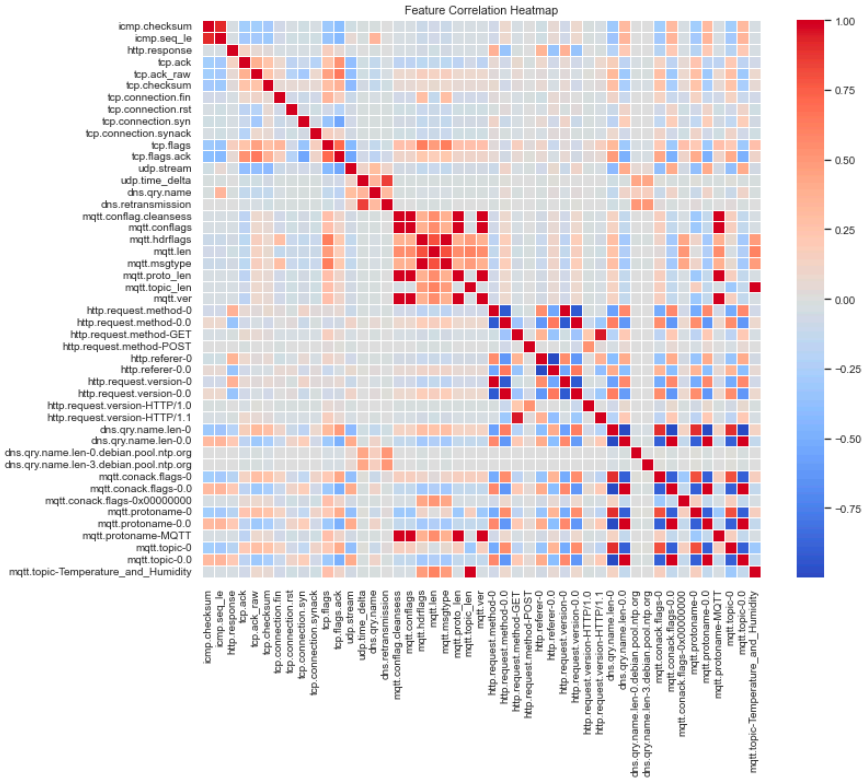

Figure 5. Feature correlation heatmap (after variance filtering).

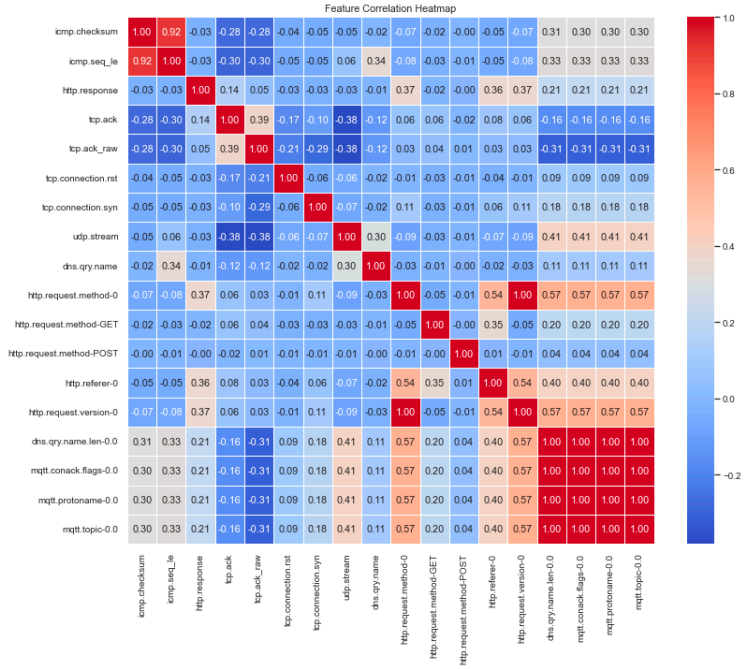

Figure 6. Feature-related heatmap (after mutual information and correlation).

 doi:10.20944/preprints202601.2418.v1

Figure 7 shows the results of the feature selection ablation experiment, where steps 1, 2, 3, and 4 correspond to the mutual information feature selection algorithm, the mRMR method, the proposed method, and all features, respectively. Except for step 4, the number of feature subsets in each step is 18. Experimental results show that the proposed hybrid feature selection algorithm outperforms the mutual information method and the mRMR method.

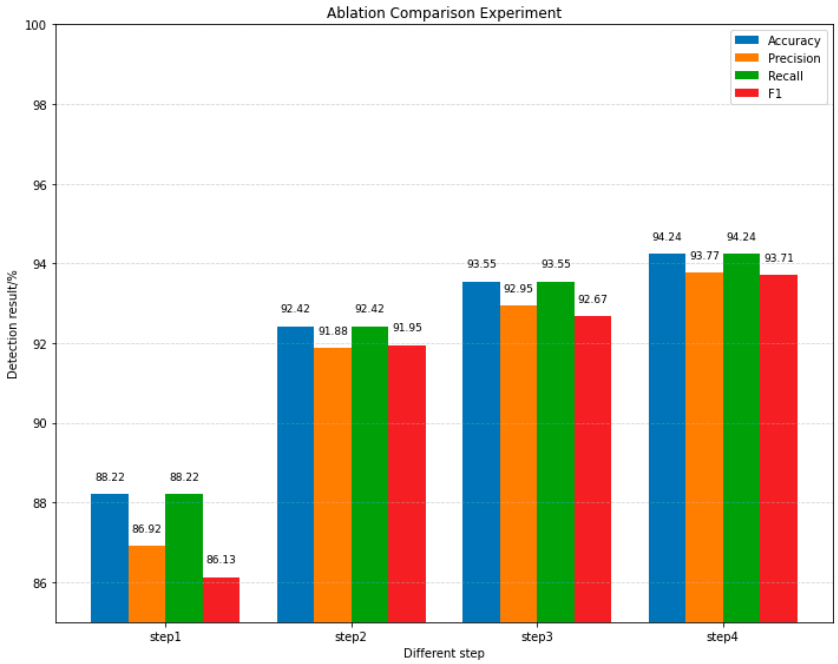

Figure 7. Comparison of ablation experiments.

4.5. TCN Training Experiment Results

Table 3 shows that the TCN training model achieves the best accuracy, recall, and F1 score when the expansion factor d is 8 and the kernel size w is 1. As the expansion factor d increases, the number of layers and parameters in the model increases, leading to an increase in training parameters. Considering all factors, this paper selects a parameter configuration of expansion factor d =8 and kernel size w =1.

Table 3. Test results for different setting of parameters d and w.

d	w	Accuracy /%	Precision /%	Recall /%	F1 /%	Model parameters /KB
2	1	92.18	92.74	93.18	91.71	9
2	2	93.20	92.20	93.20	92.21	13
2	4	93.19	92.13	93.19	92.29	21
4	1	93.14	93.02	93.14	91.69	12.37

 doi:10.20944/preprints202601.2418.v1

4	2	93.14	92.62	93.14	92.16	18.37
4	4	93.12	92.60	93.12	91.69	30.37
8	1	93.55	92.95	93.55	92.67	15.75
8	2	93.24	92.51	93.24	92.43	23.75
8	4	93.16	92.08	93.16	92.27	39.75
16	1	93.25	92.56	93.25	92.21	19.12
16	2	93.28	92.52	93.28	92.48	29.12
16	4	93.21	92.42	93.21	92.35	49.12

To verify the detection performance of the TCN model, this paper conducted comparative experiments with RNN, LSTM, and CNN-LSTM intrusion detection methods, and the results are shown in Table 4. As can be seen from the table, the proposed TCN model outperforms RNN, LSTM, and CNN-LSTM models in terms of detection performance. Furthermore, the TCN model has only 15.75 KB of parameters, which is lower than the other models, demonstrating a significant reduction in model complexity while maintaining detection accuracy. Therefore, from the perspective of combining detection performance and lightweight design, the TCN method exhibits optimal performance in intrusion detection tasks and is more suitable for deployment in edge environments with limited computing resources.

Table 4. Comparison of performance of different method.

Model	Accuracy /%	Precision /%	Recall /%	F1 /%	Model parameters /KB
RNN	92.80	91.31	92.80	91.80	24.56
LSTM	91.40	89.39	91.40	89.68	27.43
CNN- LSTM	92.71	91.13	92.71	91.57	55.06
TCN	93.55	92.95	93.55	92.67	15.75

Figures 8 and 9 show the accuracy and loss values of the four detection models on the validation set as a function of epochs. As can be seen from the figures, the TCN model converges significantly faster than the other models. This is because TCN achieves efficient sequence modeling through dilated causal convolution, effectively alleviating the gradient vanishing problem by combining residual connections, thereby improving the model's training efficiency and stability.

 doi:10.20944/preprints202601.2418.v1

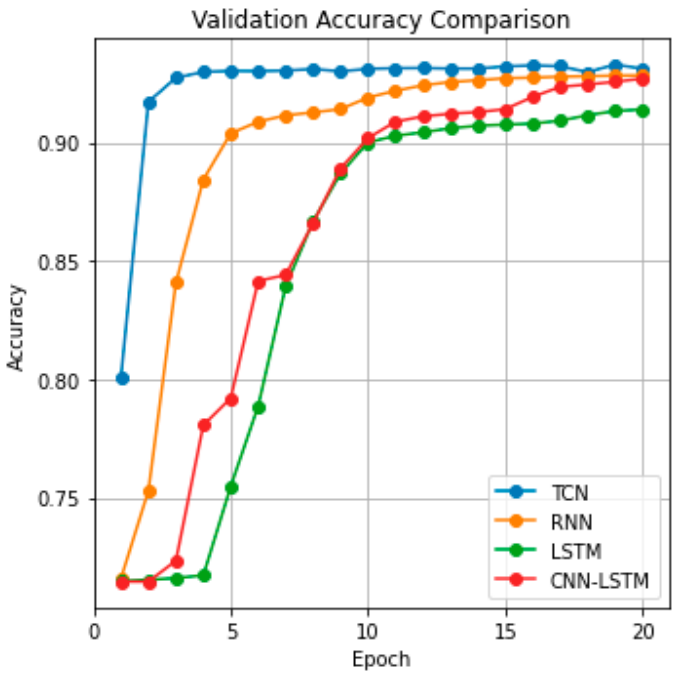

Figure 8. Comparison of Validation Accuracy Across Different Models.

As illustrated in Figure 10, the model can accurately distinguish most types of intrusion attacks, and performs particularly well in the detection of typical DDoS attacks, MITM attacks and normal traffic. Although there is still some confusion in some attack types with similar categories and a small number of samples. However, upon considering other performance metrics, the model is robust.

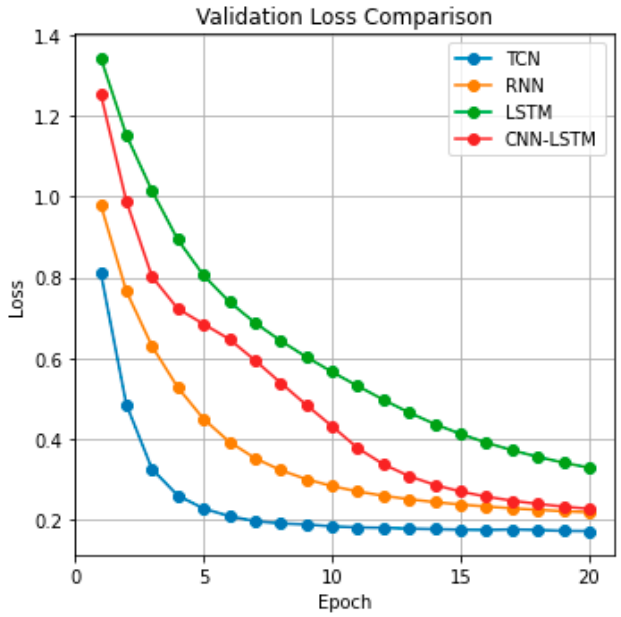

Figure 9. Comparison of Validation Loss Across Different Models.

 doi:10.20944/preprints202601.2418.v1

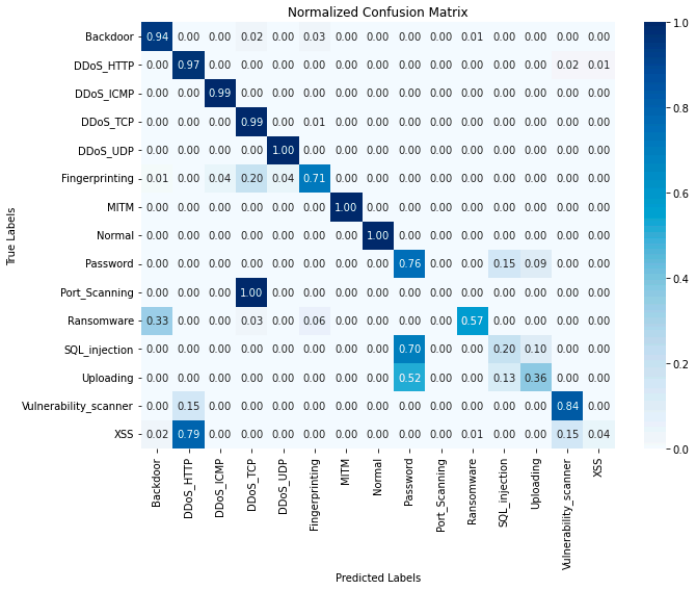

Figure 10. Confusion Matrix of Classification on the Edge-IIoTset Dataset.

5. Conclusions

This paper addresses the challenges of high-dimensionality and severe feature redundancy in network traffic data within the IoT environment, as well as the high computational complexity and difficulty in deploying existing intrusion detection models on edge devices. A hybrid feature selection method combined with a lightweight deep learning intrusion detection model is proposed. In the hybrid feature selection stage, variance filtering, mutual information, and Pearson Correlation Coefficient are integrated to effectively reduce feature dimensionality and redundancy, providing a subset of features with strong discriminative power and low feature dimensionality for subsequent model training. Experimental results show that even when the feature dimension is compressed from 94 to 18, the proposed feature selection method still exhibits good detection performance. Based on this, a lightweight intrusion detection model based on TCN is constructed and validated on the Edge-IIoTset dataset. Experimental results demonstrate that the TCN model has significant advantages in detection performance ,providing a new approach for deploying intrusion detection system at the edge. In future work, we will investigate the impact of data imbalance to feature selection and TCN model, thereby further enhancing the applicability and robustness of the model for anomalous IoT traffic prediction.

Author Contributions: Conceptualization, W.G. and M.W.; methodology, W.G. and F.L.; software, W.G. and Y.P.; validation, Y.P., C.W., and W.G.; formal analysis, W.G.; investigation, M.W.; resources, C.W.; data curation, C.W.; writing—original draft preparation, W.G.; writing—review and editing, C.W. and M.W.; visualization, Y.P.; supervision, F.L.; project administration, F.L.; funding acquisition, M.W. All authors have read and agreed to the published ver-sion of the manuscript.

Funding: This research was funded by the Major Project of Science and Technology Research Program of Chongqing Municipal Education Commission of China (Grant number KJZD-K202302401), the Youth Project of Science and Technology Research Program of Chongqing Education Commission of China (Grant number KJQN202302402), and the Applied Research Project of Chongqing College of Mobile Communication (Grant number KY20240007).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

 doi:10.20944/preprints202601.2418.v1

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: This work was supported by the Chongqing Key Laboratory of Public Big Data Security Technology.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Alotaibi B. A survey on industrial Internet of Things security: Requirements, attacks, AI-based solutions, and edge computing opportunities[J]. Sensors, 2023, 23(17): 7470.
2. Kumar M, Kumar A, Verma S, et al. Healthcare Internet of Things (H-IoT): Current trends, future prospects, applications, challenges, and security issues[J]. Electronics, 2023, 12(9): 2050.
3. Omrany H, Al-Obaidi K M, Hossain M, et al. IoT-enabled smart cities: a hybrid systematic analysis of key research areas, challenges, and recommendations for future direction[J]. Discover Cities, 2024, 1(1): 2.
4. Popoola O, Rodrigues M, Marchang J, et al. A critical literature review of security and privacy in smart home healthcare schemes adopting IoT & blockchain: problems, challenges and solutions[J]. Blockchain: Research and Applications, 2024, 5(2): 100178.
5. Kumar S, Kumar D, Dangi R, et al. A review of lightweight security and privacy for resource-constrained IoT devices[J]. Computers, Materials and Continua, 2024, 78(1): 31-63.
6. Chen K, Zhang S, Li Z, et al. Internet-of-things security and vulnerabilities: Taxonomy, challenges, and practice[J]. Journal of Hardware and Systems Security, 2018, 2(2): 97-110.
7. Khraisat A, Gondal I, Vamplew P, et al. Survey of intrusion detection systems: techniques, datasets and challenges[J]. Cybersecurity, 2019, 2(1): 1-22.
8. Ahmad M, Riaz Q, Zeeshan M, et al. Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set[J]. EURASIP Journal on Wireless Communications and Networking, 2021, 2021(1):
9. Chinnasamy R, Subramanian M, Easwaramoorthy S V, et al. Deep learning-driven methods for network-based intrusion detection systems: A systematic review[J]. ICT Express, 2025.
10. Gyamfi E, Jurcut A. Intrusion detection in internet of things systems: a review on design approaches leveraging multi-access edge computing, machine learning, and datasets[J]. Sensors, 2022, 22(10): 3744.
11. Pudjihartono N, Fadason T, Kempa-Liehr A W, et al. A review of feature selection methods for machine learning-based disease risk prediction[J]. Frontiers in bioinformatics, 2022, 2: 927312.
12. Omuya E O, Okeyo G O, Kimwele M W. Feature selection for classification using principal component analysis and information gain[J]. Expert Systems with Applications, 2021, 174: 114765.
13. Alhassan S, Abdul-Salaam G, Micheal A, et al. CFS-AE: Correlation-based Feature Selection and Autoencoder for Improved Intrusion Detection System Performance[J]. J. Internet Serv. Inf. Secur., 2024, 14(1): 104-120.
14. Devaraju S, Ramakrishnan S, Jawahar S, et al. Entropy-based feature selection for network intrusion detection systems[M]//Methods, Implementation, and Application of Cyber Security Intelligence and Analytics. IGI Global Scientific Publishing, 2022: 201-225.
15. Xie S, Zhang Y, Lv D, et al. A new improved maximal relevance and minimal redundancy method based on feature subset[J]. The Journal of supercomputing, 2022, 79(3): 3157.
16. Rahman M M, Al Shakil S, Mustakim M R. A survey on intrusion detection system in IoT networks[J]. Cyber Security and Applications, 2025, 3: 100082.
17. Huang H, Wang P, Pei J, et al. Deep learning advancements in anomaly detection: A comprehensive survey[J]. IEEE Internet of Things Journal, 2025.
18. Ullah I, Mahmoud Q H. Design and development of RNN anomaly detection model for IoT networks[J]. IEEE Access, 2022, 10: 62722-62750.

 doi:10.20944/preprints202601.2418.v1

19. Elshewey A M, Abbas S, Osman A M, et al. DDoS classification of network traffic in software defined networking SDN using a hybrid convolutional and gated recurrent neural network[J]. Scientific Reports, 2025, 15(1): 29122.
20. Kodyš M, Lu Z, Fok K W, et al. Intrusion detection in internet of things using convolutional neural networks[C]//2021 18th international conference on privacy, security and trust (PST). IEEE, 2021: 1-10.
21. Lopes I O, Zou D, Abdulqadder I H, et al. Network intrusion detection based on the temporal convolutional model[J]. Computers & Security, 2023, 135: 103465.
22. Nazre R, Budke R, Oak O, et al. A temporal convolutional network-based approach for network intrusion detection[C]//2024 International Conference on Integrated Intelligence and Communication Systems (ICIICS). IEEE, 2024: 1-6.
23. He P, Zhang H, Feng Y, et al. A design of network attack detection using causal and non-causal temporal convolutional network[C]//International Conference on Science of Cyber Security. Cham: Springer Nature Switzerland, 2023: 513-523.
24. Ferrag M A, Friha O, Hamouda D, et al. Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning[J]. IEEe Access, 2022, 10: 40281-40306.
25. Latif S, Boulila W, Koubaa A, et al. Dtl-ids: An optimized intrusion detection framework using deep transfer learning and genetic algorithm[J]. Journal of Network and Computer Applications, 2024, 221: 103784.
26. Peng H, Long F, Ding C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy[J]. IEEE Transactions on pattern analysis and machine intelligence, 2005, 27(8): 1226-1238.
27. Ambusaidi M A, He X, Nanda P, et al. Building an intrusion detection system using a filter-based feature selection algorithm[J]. IEEE transactions on computers, 2016, 65(10): 2986-2998.