

Received 19 March 2024, accepted 3 April 2024, date of publication 8 April 2024, date of current version 22 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3386405

## RESEARCH ARTICLE

# BBO-CFAT: Network Intrusion Detection Model Based on BBO Algorithm and Hierarchical Transformer

TINGYAO JIANG<sup>1</sup>, XIAOBO FU<sup>1</sup>, AND MIN WANG<sup>2</sup>

<sup>1</sup>College of Computer and Information Technology, China Three Gorges University, Yichang 443005, China

<sup>2</sup>School of Electronic Information, Hubei Three Gorges Polytechnic, Yichang 443002, China

Corresponding author: Xiaobo Fu (202108120021004@ctgu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61872221.

**ABSTRACT** In today's network environments, vulnerable to cyber threats such as hackers and viruses, intrusion detection technology is considered the most effective means of detection and defense. Deep neural networks are commonly used in intrusion detection technology. However, improving the model's ability to extract feature information and reducing computational space while retaining local feature information are critical challenges that need to be addressed. To tackle these issues, this paper proposes a model named BBO-CFAT, which combines the Biogeography-Based Optimization algorithm (BBO) for feature selection and an improved Transformer model for preserving context information and reducing computational space. Specifically, the BBO-CFAT model employs a roulette selection method to control the operations of migration and mutation operators. It utilizes feature information entropy to weight updates of adaptive variables in these operators, thereby enhancing the credibility of feature selection. Furthermore, the Transformer framework is hierarchically designed to facilitate the acquisition of context information. Additionally, depthwise separable convolutions are employed to reduce computational space, thereby improving computational efficiency and training speed. Experimental evaluations using the CIC-IDS2017 and NSL-KDD datasets demonstrate promising accuracies for BBO-CFAT on both datasets, achieving 99.1% and 97.5% accuracy, respectively, surpassing the performance of comparative experiments. Overall, the BBO-CFAT model provides a comprehensive solution to the challenges of intrusion detection, effectively balancing feature preservation, computational efficiency, and training accuracy.


**INDEX TERMS** Intrusion detection, BBO, transformer, feature selection.

## I. INTRODUCTION

In recent years, there has been rapid development in computer and Internet technology. Internet technology has become an indispensable element of people's daily lives, facilitating online payment, ticket and hotel room bookings, and network communication, fundamentally transforming the way people live. However, during the early stages, cyberspace lacked robust establishment, as evidenced in prior studies [1], creating favorable conditions for illicit intrusion. Ensuring timely detection of network intrusion is crucial for safeguarding digital infrastructure, mitigating

economic losses, and maintaining overall security. While traditional intrusion defense techniques, such as firewalls, identity security authentication, and access control, can offer protection against specific intrusion methods [2], [3], intrusion detection techniques (IDS) are considered the most effective in detecting attacks and ensuring network security. IDS can be categorized into four main types: network-based intrusion detection systems (NIDS), host-based intrusion detection systems (HIDS), signature-based intrusion detection systems (SIDS), and anomaly-based intrusion detection systems (AIDS).

Network intrusion detection technology is the main focus of this paper, with NIDS primarily detecting network traffic to determine attack paths [4]. Because machine learning and

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Huang<sup>1</sup>.

deep learning can leverage big data to learn and predict specific functionalities, advancements in the field of machine learning and deep learning technologies have been introduced into the domain of network intrusion detection. Common machine learning methods such as Random Forest (RF) [5], Decision Tree (DT) [6], Naive Bayes (NB) [7], and Support Vector Machine (SVM) [8] have been used to create NIDS models. In contrast, deep learning models with deeper and more extensive structures demonstrate higher effectiveness in feature extraction. For instance, Longaris et al. [9] introduced an autoencoder-based Long Short-Term Memory (LSTM) model to identify anomalous behavior in networks. On the other hand, Fernandez et al. [10] proposed a model based on Deep Neural Networks (DNN) and tested its robustness. Elnakib et al. [47] introduced an Enhanced Intrusion Detection Model (EIDM) using deep learning techniques, which effectively identified unauthorized access, anomalies, and malicious activities in Internet of Things (IoT) networks through attention mechanisms or transfer learning. Choobdar et al. [48] addressed intrusion detection and multi-class classification issues in software-defined networks by extracting useful features from network traffic data using stacked autoencoder technology. This method enables effective differentiation between normal traffic and malicious behavior. Olimov et al. [54] proposed a novel unsupervised learning method based on graph Laplacian matrix for anomaly detection and localization. This method effectively detects and localizes anomalies using graph structural information without relying on labeled anomaly data. It demonstrates outstanding performance and accuracy across various datasets, highlighting its effectiveness.

This paper primarily investigates intrusion detection technology using deep neural networks, which, when combined with hybrid models and increased network depth to enhance detection capabilities, also expose certain issues. These include high redundancy and high-dimensional imbalances in the data, optimization of how models preserve local feature extraction and computational efficiency, and ensuring accurate completion of feature selection to ensure the importance of selected features.

To address these issues, we have proposed a series of methods and techniques. We introduce an enhanced feature selection method designed to preserve crucial features. Second, our improved Transformer model not only advances feature information extraction but also takes into account global feature information and contextual information between layers, leading to improved computational efficiency.

The paper is organized as follows: In Section II, we provide an overview of the background knowledge related to feature selection and intrusion detection models as presented in the existing literature. Section III elaborates on the proposed method and its associated parameters. Section IV offers insights into the experimental results. Finally, in Section V,

we summarize the contributions of this paper and outline potential directions for future research.

## II. RELATED WORK

This section summarizes the algorithms and research relevant to this study.

### A. FEATURE SELECTION

Feature selection often directly impacts the model's effectiveness, so it is necessary to perform feature selection prior to network training to ensure global optimality during training. Mafarja et al. [11] proposed an improved feature selection method based on the Whale Optimization Algorithm (WOA). The method introduced the V-shaped and S-shaped methods to the WOA algorithm for improved feature selection while dealing with high dimensionality. Alamiedy et al. [12] proposed an improved intrusion detection model based on multi-objective grey wolf optimization, which uses GWO as a method to obtain the most practical features and support vector machine to predict Dos attack, Probe attack, R2L attack model, and U2R attack, and obtained 93.64%, 91.01%, 57.72%, 53.7% prediction results. However, the model is not balanced with a dataset, so the difference in prediction results obtained with different amounts of data is noticeable. Liu et al. [13] used Improved Social Spider Optimization (ISSO) algorithm to achieve feature extraction and selection in intrusion detection. Al-Yaseen et al. [14] proposed a feature fusion superposition integration mechanism (MFFSEM) for detecting anomalous behavior. Mainly, multiple integrated feature datasets are constructed by association and correlation between the fundamental datasets to meet the requirements of anomalous behavior detection. Halim et al. [15] proposed an enhanced feature selection method (GbFS) based on genetic algorithms to improve the accuracy of classifiers.

Simon [16] proposed a biogeographic optimization-based (BBO) algorithm, the algorithm presented in this paper is derived from the study of biological species and geographical principles, primarily aimed at addressing high-dimensional, multi-objective optimization problems. Guendouzi et al. [17] used BBO for feature selection work and verified the credibility of the feature selection work.

### B. INTRUSION DETECTION MODEL

In recent years, machine and deep learning have been widely used in intrusion detection and achieved good experimental results. Most traditional machine learning methods are based on supervised learning models [18], [19], [20]. Bhattacharya et al. [21] used Principal Component Analysis (PCA) and the Firefly algorithm to design a new classification model for intrusion detection. The model first reduces the dimensionality of the data using the PCA-Firefly algorithm and then completes the data classification using the XGBoost algorithm. Chang et al.

[22] verified that the Random Forest algorithm can be used to select the essential data features and completes the classification of the experimental data using Support Vector Machine.

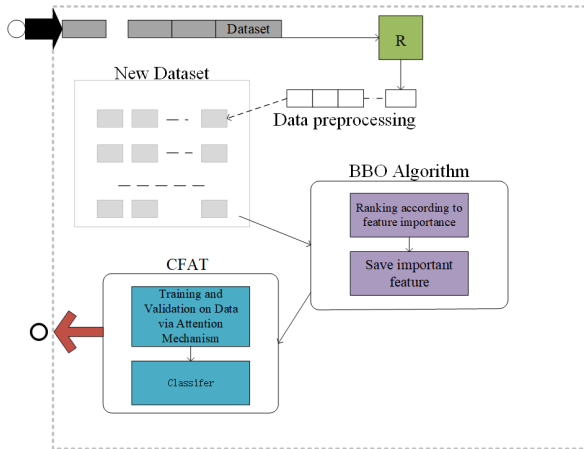
As the network data becomes high dimensional with the increasing number of users, the training accuracy of traditional machine learning decreases. Ding et al. [23] proposed a model based on KNN and Generative Adversarial Networks (TACGAN), which first uses the K-nearest neighbor method for effective downsampling, then uses TACGAN for oversampling, and finally mixes the two types of data for data balancing. Kan et al. [24] proposed an intrusion detection system based on multi-scale convolutional neural networks (CNNs) for networked communication. Furthermore, the proposed model has a faster convergence speed than AdaBoost and recurrent neural networks (RNN). Xu et al. [25] designed a deep neural network (DNN) detection framework FC-Net, which divides the model into feature extraction and comparison networks. The model learns the feature maps used for classification and then determines whether they are of the same class through the comparison network. Vaswani et al. [26] proposed Transformer, which has been rapidly applied in various fields due to its robust feature extraction capabilities. Li et al. [27] proposed combining extended convolutional neural networks with different expansion rates with Transformer to obtain coarse and fine-grained information and improve the model's generalization ability. Chen et al. [28] proposed GTA, which includes a connection learning strategy and influence propagation convolutional kernel multi-branch attention mechanism. Tuli et al. [29] proposed TranAD based on Transformer's encoder and focus score for robust multi-model feature extraction and adversarial training. Han et al. [30] proposed an intrusion detection model, GTID, which leverages N-Gram frequency for contextual information, integrates a time-aware transformer to capture the time intervals between data packets, and performs classification based on learned time features. This approach addresses the issues of information loss and feature dimensionality caused by feature extraction. Guo et al. [31] designed a CMT network with a hierarchically connected structure so that contextual hierarchical information can be retained. Wu et al. [32] designed Fastformer, which changes the multi-head attention mechanism compared to the traditional Transformer model, i.e., the traditional multi-head attention mechanism is derived by dot product, which increases the computational difficulty. The Fastformer model uses additive attention to compute attention. This makes the calculation faster and easier. Ullah et al. [33] designed a DL-based IDS to detect and classify attacks in IoT networks. The proposed convolutional neural network (CNN) model is validated using the BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, and IoT-23 intrusion detection datasets. Experimental results show that this method has high classification accuracy. However, the method uses the DL method for data training, which requires

a lot of computing resources and memory resources and is not suitable for the IoT. Rakhmonov et al. [34] firstly, generate anomaly images, and then transfer critical feature information from the network to the novice network using knowledge distillation technique, forming an end-to-end anomaly detection method, which yields impressive results. Kim et al. [35] introduced a hybrid model combining CNN and LSTM networks to extract features from real-time HTTP traffic. Through training, this model effectively differentiated complex attacks and accurately analyzed unknown web-based attacks. Alin et al. [36] proposed a hierarchical deep learning approach for intrusion detection. Henry et al. [37] The author proposes a technique that combines CNN and GRU, in which different CNN-GRU combination sequences are proposed to optimize network parameters. The results indicate significant improvements, with many network attacks detected with an accuracy of 98.73% and an FPR rate of 0.075. Altunay [38] introduced a deep learning architecture combining CNN and LSTM to detect intrusions in IoT networks, and the accurate detection success rate for attack types in the dataset was effectively evaluated. Additionally, Sharma et al. [39] proposed a filter-based feature selection deep neural network model, which removed highly correlated features and generated a few attack data using GANs to balance the dataset.

In summary, while significant research has been conducted on feature selection and deep intrusion detection models, challenges remain. Feature selection encounters issues regarding the scientific selection of feature subsets, while deep intrusion models suffer from large volume, time-consuming training, and ineffective global feature extraction. Therefore, this paper proposes enhancements through an improved BBO algorithm to enhance the scientific selection of feature subsets and an improved Transformer to retain global feature information, thereby improving computational efficiency.

### III. METHODOLOGY

This article aims to solve the problem of high latitude data, improve the scientificity of feature selection, preserve the global feature information of feature extraction, and improve computational speed. We propose a deep network intrusion detection model based on BBO feature selection method and Transformer encoder. The model consists of three stages: first, preprocessing the raw data, using unilateral selection (OSS) and boundary SMOTE to upsample and downsample the raw data, and then using WGANs network to generate a small amount of attack data; Subsequently, the improved BBO algorithm is used to rank the data features and retain the higher ranked features as features in the new dataset, which are then passed on to the training model; Finally, the designed network model is used for data classification training. The framework of the model process is shown in Figure 1:



**FIGURE 1.** The flowchart exhibits the sequence of operations, from the initial application of the BBO to the subsequent utilization of the Transformer architecture for feature selection.

**TABLE 1.** Dataset results processed by CIC-IDS2017.

Index	New labels	Old labels	old total	new total
1	Normal	Benign	2095057	199880
2	Botnet Force	FTP-PAtator,SSH-PAtator	9150	9150
3	Dos/DDos	DDos,Dos GoldenEye, Dos,Dos Hulk,Heartbleed Dos Slowhttptest Dos slowlories	321770	199667
4	Infiltration	Infiltration	36	2000
5	Portscan	PortScan	90694	90694
6	Web Attack	Web Attack-Brute Force Web Attack-Sql Injection Web Attack-XSS	2143	2143
7	Brute ARES	Bot	1948	1948

## A. DATA PREPROCESSING

The data preprocessing primarily aims to balance the dataset to enhance the intrusion detection model's ability to recognize and classify all categories, thereby improving the model's generalization and application effectiveness. In our experiments, we utilized the CIC-IDS2017 and NSL-KDD datasets, both characterized by high dimensionality and data imbalance. To mitigate the impact of the dataset on training performance, we employed a combination of one-sided selection (OSS) and boundary BiSMOTE sampling techniques to balance the dataset. This method involves undersampling to retain minority classes and reduce the quantity of majority classes, followed by oversampling using BiSMOTE with weighted settings to further sample the data. This step may introduce noise and boundary data issues. Additionally, we utilized WGANs networks to synthesize attack instances of minority classes, further balancing the dataset. This sampling approach helps avoid generating boundary data and reduces the impact of noise during the sampling process. Tables 1 and 2 display the results of processing the two datasets.

## B. IMPROVED BBO FEATURE SELECTION METHOD

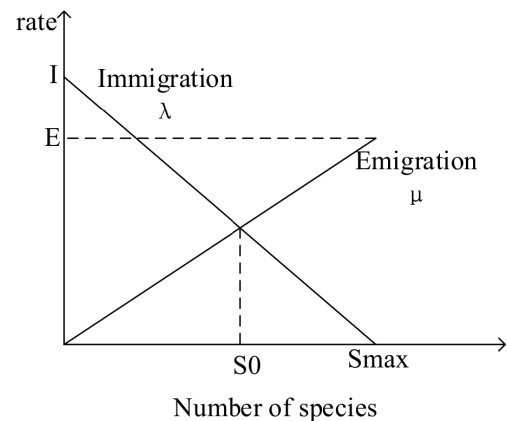
Deep learning plays a crucial role in the field of network intrusion detection technology, primarily owing to the

**TABLE 2.** Dataset results processed by NSL-KDD.

Type	old records	new records
Normal	13449	9960
Probing	2289	9804
R2L	209	6000
U2R	11	2500
Dos	9234	9234

multi-layered neural network models' ability to capture extensive feature information and automatically conduct feature learning. However, efficiently utilizing neural networks to focus on significant data features for feature learning can enhance testing accuracy. Therefore, this study proposes an improved BBO algorithm for data feature selection.

In the BBO algorithm, species can migrate between islands, with each island termed a habitat. Each habitat represents a species' range and is characterized by its Habitat Suitability Index (HSI), which indicates the quality of life within the habitat. HSI is determined by Suitability Index Variables (SIV), influenced by specific habitat conditions such as rainfall, species diversity, and habitat size. Habitats with high HSI values feature high species saturation, displaying low immigration rates and high emigration rates, whereas habitats with low HSI values often harbor fewer species, leading to low emigration rates and high immigration rates. Figure 2 illustrates data related to habitats with immigration and emigration rates.



**FIGURE 2.** Distribution of species in the habitat.

In this algorithm, each data category in the original dataset is treated as an independent habitat, represented by a binary carrier with SIV (Suitability Index Variables), and the initial feature values are set to all zeros to signify the transformation. Subsequently, migration and mutation rates are computed for each habitat, and migration or mutation thresholds are established. During migration or mutation operations, instead of comparing the migration or mutation rates of each habitat through traditional random number generation, the entropy of the habitat to be migrated or mutated is calculated as a weight against the migration or mutation rate. Then, this weight is compared with the initial habitat threshold using a roulette selection method, and each habitat is updated after each



migration or mutation operation, ensuring that each update considers the information of the population. The specific operations are as follows: when performing migration and mutation operations, we need to use the roulette selection method to select the  $SIV[i]$  to be operated on, obtain its entropy value, and compare it with the migration rate. The calculation method for entropy is shown in Formula 1:

$$prob[i] = - \sum P(x_i) \log P(x_i) \quad (1)$$

**TABLE 3.** CIC-IDS2017 feature subset filtered by BBO.

Ranks	Features	Ranks	Features
1	ECE Flag Count	31	Fwd Packet Length Min
2	RST Flag Count	32	Init_Win_bytes_forward
3	Fwd Avg Packets/Bulk	33	Fwd IAT Min
4	Bwd Avg Bulk Rate	34	Packet Length Variance
5	Bwd URG Flags	35	Init_Win_bytes_backward
6	Bwd Avg Bytes/Bulk	36	Active Max
7	Fwd URG Flags	37	Destination Port
8	Bwd Avg Packets/Bulk	38	Min Packet Length
9	Fwd Avg Bulk Rate	39	Total Fwd Packets
10	CWE Flag Count	40	Flow Duration
11	Fwd Avg Bytes/Bulk	41	Bwd IAT Min
12	Bwd PSH Flags	42	Total Length of Bwd Packets
13	Fwd PSH Flags	43	Flow IAT Min
14	SYN Flag Count	44	Fwd Packet Length Std
15	URG Flag Count	45	Total Backward Packets
16	FIN Flag Count	46	Avg Bwd Segment Size
17	Active Std	47	Total Length of Fwd Packets
18	Fwd Packet Length Min	48	Average Packet Size
19	ACK Flag Count	49	Fwd IAT Mean
20	Subflow Fwd Bytes	50	Bwd Packet Length Min
21	Fwd IAT Total	51	Active Min
22	Active Mean	52	Bwd IAT Std
23	Flow IAT Std	53	Idle Mean
24	Bwd Packets/s	54	Fwd Packet Length Mean
25	Max Packet Length	55	Packet Length Mean
26	Flow IAT Max	56	PSH Flag Count
27	Fwd IAT Std	57	Subflow Fwd Packets
28	Idle Max	58	Bwd Packet Length Std
29	Flow IAT Mean	59	Down/Up Ratio
30	Bwd Packet Length Max	60	Fwd Packets/s

**TABLE 4.** NSL-KDD feature subset filtered by BBO.

Ranks	Features	Ranks	Features
1	src_bytes	7	dst_host_same_src_port_rate
2	dst_bytes	8	dst_host_srv_diff_host_rate
3	dst_host_count	9	dst_host_serror_rate
4	dst_host_srv_count	10	dst_host_srv_serror_rate
5	dst_host_same_srv_rate	11	dst_host_rerror_rate
6	dst_host_diff_srv_rate	12	dst_host_srv_rerror_rate

where  $x_i \in \mathbb{R}_{m \times n}$  represents the  $i$ -th attribute value,  $P(x_i)$  denotes the probability of the  $i$ -th SIV, then using the information entropy as weights to update the value of SIVs[i], the specific calculation is as shown in Formula 2:

$$SIV'[k] = SIV[k] + prob[i] * (SIV[i] - SIV[k]) \quad (2)$$

where  $i$  is the information entropy of the  $i$ -th feature after roulette selection, prob is the information entropy of the  $i$ -th feature,  $SIV[i]$  is the most weighted of the SIVs, and  $SIV[j]$  is the least weighted of the SIVs. After several rounds of selection, the performance of the SIVs is obtained.

The distribution of features can be visualized, and the final ranking of features in the CIC-IDS 2017 dataset and the NSL-KDD can be derived as shown in Table 3 and Table 4:

### C. CFAT

After feature selection, the dataset retains the most crucial feature values, enabling subsequent model training to focus on these key features. To preserve global feature information and further enhance the model's computational efficiency, we propose the CFAT model. This model redesigns the network architecture into a hierarchical structure, connecting various layers and employing CNN networks between layers to facilitate information flow. Additionally, it incorporates the LPU module into the multi-head attention mechanism, aiming to preserve contextual information and establish residual connections with the results of using linear addition to compute attention vectors, promoting the fusion of features into global vectors. To further reduce computational complexity, the multi-head attention mechanism utilizes deep separable convolutional networks to decrease computational space and improve computation speed. The network model of CFAT is illustrated in Figure 3.

In the traditional Transformer network, given an input of size  $R \times m \times n$ , the original multi-head attention mechanism first generates Query, Key, and Value matrices. By performing dot product operations between Query and Key, it produces a weight matrix of size  $R \times m \times n$ , as shown in Equation (3).

$$Attn(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (3)$$

where  $d_k$  is the number of columns of the Q,K matrix, i.e., the vector dimension.

In this paper, we first utilize a new dataset  $E \in \mathbb{R}^{N \times d}$ , where  $N$  is the query length and  $d$  is the dimensionality obtained after feature selection. This dataset is transformed into linear sequences  $Q, K$  and  $V$  through linear transformations. The dimensions of the generated  $Q, K$  and  $V$  matrices are  $Q, K, V \in \mathbb{R}^{N \times d}$ , where  $Q = [q_1, q_2, q_3, \dots, q_d]$ ,  $K = [k_1, k_2, k_3, \dots, k_d]$ , and  $V = [v_1, v_2, v_3, \dots, v_d]$ . To enhance the interaction efficiency between data queries, keys, and values, and to reduce the complexity of computing attention, we adopt an additive attention mechanism. This mechanism enables retrieval of crucial information in linear complexity. Therefore, we first use the additive attention mechanism to transform queries, keys, and values into  $q, k$ , and  $v$  respectively, each of dimension  $q, k, v \in \mathbb{R}_d$ , to retain contextual information. Specifically, the attention weight for the  $i$ -th query vector is computed as shown in Equation (4).

$$\alpha_i = \frac{\exp(W_q^T q_i / \sqrt{d})}{\sum_{j=1}^N \exp(W_q^T q_j / \sqrt{d})} \quad (4)$$

where  $W_q \in \mathbb{R}_d$  is a learnable parameter vector, thus the global query vector is computed as shown in Equation (5).

$$q = \sum_{i=1}^N \alpha_i q_i \quad (5)$$

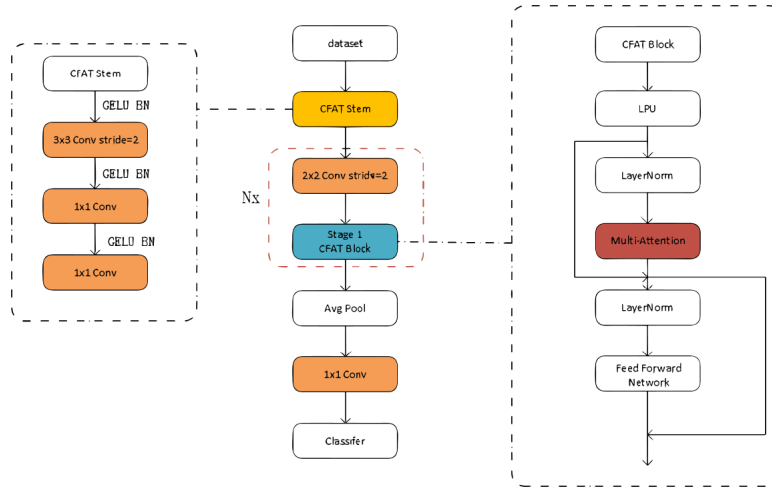


FIGURE 3. CFAT network model.

In the subsequent steps, the obtained global vector interacts with the Key matrix to retain the contextual information from the previous part. Thus, a global vector “k” containing context information is obtained. The  $i$ -th variable in this matrix is denoted as  $p_i = q \cdot K_i$ , allowing for the calculation of the attention weight for the  $i$ -th key vector, as shown in Equation (6).

$$\beta_i = \frac{\exp(W_{k'}^T p_i / \sqrt{d_k})}{\sum_{j=1}^N \exp(W_{k'}^T p_j / \sqrt{d_k})} \quad (6)$$

where  $W_{k'} \in \mathbb{R}_d$  is a learnable parameter vector, thus the global query vector is as shown in Equation (7).

$$k = \sum_{i=1}^N \beta_i p_i \quad (7)$$

By applying depth separable convolution with a stride of  $K'$ , represented as  $DWConv(K)$ , where  $K' \in \mathbb{R}^{(N/k^2 * d_k)}$ , the computational space is compressed by a factor of  $k$ , resulting in the acceleration of the calculation of the K-V interaction vector. This approach underscores the advantages of utilizing depth separable convolution, which significantly enhances the computational efficiency of the K-V interaction vector, denoted as  $u_i$ .

Similarly, vector weights for the  $i$ -th value vector can be obtained, and computational space can be compressed using  $k \times k$  depth-separable convolution, as shown in Equation 8.

$$\gamma_i = \frac{\exp(W_{v'}^T t_i / \sqrt{d_v})}{\sum_{j=1}^N \exp(W_{v'}^T t_j / \sqrt{d_v})} \quad (8)$$

where  $W_{v'} \in \mathbb{R}_d$  is a learnable parameter vector, and  $t_i$  is obtained by taking the dot product of the global variable  $k$  and  $V_i$ . Thus, the global query vector is as shown in Equations 9 and 10:

$$v = \sum_{i=1}^N \gamma_i t_i \quad (9)$$

$$V' = DWConv(V) \in \mathbb{R}^{\frac{N}{\sqrt{2}} * d_v} \quad (10)$$

Then the K-V interaction vector  $u_i$  is calculated with the expression  $u_i = k * v_i$ . Finally, the output is obtained by summing with the original query vector. The structure is shown in Figure 4.

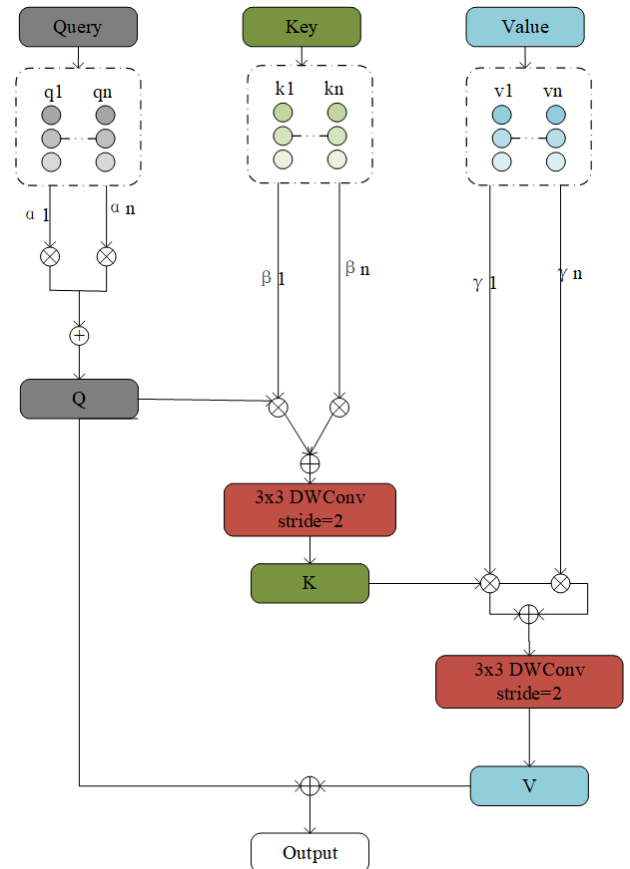


FIGURE 4. The data is first transformed into linear vectors and context information is compressed into global vectors. Subsequently, key-value pairs are compressed using depth separable convolution to reduce computational space.

**TABLE 5.** Temperature and wildlife count in the three areas covered by the study.

Real \ Projected	Positive	Negative
Positive	TP	FN
Negative	FP	TN

## IV. EXPERIMENTS

### A. EVALUATION METRICS

The experiment is calculated from Table 5.

The True Positive (TP) samples are those with a positive true category, and the model correctly predicts them as positive. True Negative (TN) samples are those with a negative true category, and the model correctly predicts them as negative.

The experiment employs various evaluation metrics, namely Accuracy, Precision, Recall, and F1-score. Accuracy denotes the model's capability to correctly classify samples in relation to the overall sample size. Precision reflects the ratio of true positive predictions to the total number of samples predicted as positive. Recall indicates the proportion of true positive samples to the total samples that should have been accurately classified as positive. The F1-score is a harmonized mean that balances Precision and Recall. The formula for each evaluation metric is as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (11)$$

$$Precision = \frac{TP}{TP + FN} \quad (12)$$

$$Recall = \frac{TP}{TP + FP} \quad (13)$$

$$F1score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (14)$$

In the multiclass task, the accuracy of the multiclassification is obtained by dividing the number of all correctly classified data by the total number of data, shown in (14). Precision, Recall, and F1 Score are calculated separately for each category, and then these values are summed and averaged to obtain the evaluation criteria for the multiclassification task. For example, Precision is calculated as shown in (15):

$$Accuracy_{multi} = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n (TP_i) + FP_i} \quad (15)$$

$$Precision_{multi} = \frac{\sum_{i=1}^n P_i}{n} \quad (16)$$

### B. PARAMETER SETTINGS

The difference in hyperparameters affects the model's convergence speed and the experiment's accuracy. The hyperparameter settings for this experiment are shown in Table 6 below:

In this experiment, the CFAT model employs the Adam optimizer to optimize model parameters, adjusting the learning rate, and utilizes CrossEntropyLoss as the loss function.

**TABLE 6.** Table of hyperparameter settings.

Parameters	Values
BatchSize	512
Drop	0.1
qkv_bias	True
ratio_ffn	3.6
input_channels	3
d_models	[256,64,128,256,512]
num_heads	1,2,4,8
rs_ratio	8,4,2,1
epoch	50

**TABLE 7.** Comparative feature selection experiments based on the CIC-IDS2017 dataset.

Reference	Method	Accuracy(%)
Zegarra et al [41]	TabNet	97.03
Yao R et al [42]	XGBoost	91.06
Kareem et al [43]	GTO-BSA	98.79
Jeyaselvi, M. et al [44]	EXPSO-STFA	95.65
Proposed	BBO	99.1

**TABLE 8.** Comparative feature selection experiments based on the NSL-KDD dataset.

Reference	Method	Accuracy(%)
Yaseen et al [14]	differential evaluation	80.15
Yao R et al [42]	XGBoost	91.06
Kareem et al [43]	GTO-BSA	95.59
Ingre et al [45]	CFS+ANN	79.9
Proposed	BBO	97.5

### C. FEATURE SELECTION EXPERIMENT AND ANALYSIS

In this section, we compare the BBO feature selection method with the latest feature selection methods. Our aim is to illustrate their similarities, highlight the unique advantages offered by our method, and reflect the algorithm's superiority through accuracy. The experimental results are shown in Table 7 and Table 8.

The experimental results presented in Tables 7 and 8 demonstrate that the proposed feature selection method achieved significant improvements in accuracy on both datasets, outperforming other comparative experiments. This suggests the feasibility and superiority of our proposed algorithm in feature selection compared to some existing methods. Particularly notable improvements were observed, especially on the NSL-KDD dataset.

### D. EXPERIMENTAL RESULTS AND ANALYSIS

To evaluate the feasibility of the proposed model on the datasets and assess its robustness in real-world scenarios, we conducted anomaly detection and prediction experiments on two datasets. In this study, we explored the superiority of the model by combining machine learning with deep learning models, including common deep composite models such as BiLSTM-DNN (BD), MultiAttention-BiLSTM-DNN (MBD), Transformer-DNN (TD), Position-Transformer-DNN (PTD), and Encoder-BiLSTM-DNN (EBD). To further validate the feasibility of the proposed model, recent intrusion detection models were

**TABLE 9.** Comparison of experimental results of different algorithms on the CIC-IDS2017 dataset.

Model	classification indicators%			
	Accuracy	Precision	Recall	F1-score
SVM	0.660	0.401	0.401	0.368
LR	0.685	0.715	0.423	0.418
NB	0.346	0.299	0.611	0.199
BD	0.980	0.753	0.793	0.767
MBD	0.650	0.539	0.551	0.545
TD	0.986	0.966	0.904	0.927
PTD	0.963	0.829	0.708	0.734
EBD	0.956	0.643	0.553	0.551
CFAT	0.991	0.972	0.941	0.945

**TABLE 10.** Comparison of experimental results of different algorithms on the NSL-KDD dataset.

Model	classification indicators%			
	Accuracy	Precision	Recall	F1-score
SVM	0.767	0.865	0.612	0.749
LR	0.778	0.868	0.631	0.764
NB	0.807	0.875	0.580	0.821
BD	0.864	0.876	0.845	0.882
MBD	0.875	0.882	0.901	0.710
TD	0.851	0.955	0.762	0.847
PTD	0.875	0.929	0.861	0.876
EBD	0.879	0.950	0.841	0.887
CFAT	0.975	0.945	0.948	0.921

selected for comparison. This experiment was implemented using PyTorch in Python and executed on a computer equipped with an NVIDIA GeForce RTX 3080Ti graphics card. The experimental results of different models on the two datasets are shown in Tables 9, Tables 10.

The above table demonstrates that the proposed model performs well on various performance metrics, surpassing most of the control experiments. It is noteworthy that the activation function Wib-ReLU, proposed by Olimov et al. [40], was employed in this experiment. This activation function addresses the issue of increasing activation mean through weight initialization, thereby facilitating smoother model training. Figures 5 and 6 depict the performance of each model in training and testing on two datasets.

**TABLE 11.** Below is a comparison of CFAT with other intrusion detection models on the CIC-IDS2017 dataset.

Reference	Method	Accaury(%)
Mehedi et al [46]	LeNet	98.10
Elnakib et al [47]	EIDM	95.00
Choobdar et al [48]	DT	98.38
Jaradat et al [49]	DT	98.50
Ours	CFAT	99.10

Furthermore, for a better assessment of the model's predictive accuracy, reference can be made to the confusion matrices depicted in Figures 7 and 8. To further validate the feasibility of the proposed model, a comparison is conducted with prior studies. As shown in Table 11 and Table 12.

**TABLE 12.** Below is a comparison of CFAT with other intrusion detection models on the NSL-KDD dataset.

Reference	Method	Accaury(%)
Vinayakumar et al [50]	DNN	78.50
Gamage et al [51]	LSTM	77.26
Alazab et al [52]	CosSimMFO+DT	89.70
Jie et al et al [53]	SingleSVM	97.39
Ours	CFAT	97.5

**TABLE 13.** Performance gains obtained for different modules on CIC-IDS2017 dataset, ✓ indicates that the corresponding module has been added to the method.

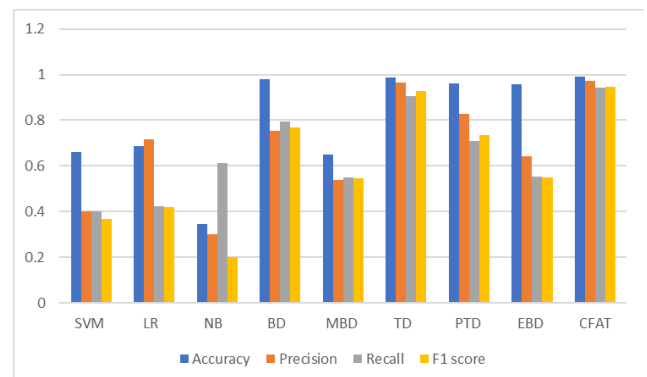
Transformer	DWconv	ReLU	wib-ReLU	Accaury(%)
✓				96.5
✓	✓			97.6
✓	✓	✓		97.5
✓	✓		✓	99.1

**TABLE 14.** Performance gains obtained for different modules on NSL-KDD dataset, ✓ indicates that the corresponding module has been added to the method.

Transformer	DWconv	ReLU	wib-ReLU	Accaury(%)
✓				90.2
✓	✓			93.2
✓	✓	✓		84.5
✓	✓		✓	97.5

## E. ABLATION EXPERIMENTS

Tables 13 and 14 summarize the performance of different modules on the two datasets. The experimental results demonstrate that each module is compatible, and the combination effect yields the best results. Additionally, the data in the tables indicate that the performance of the Wib-ReLU activation function [40] is superior to ReLU. This activation function addresses the issue of increasing the activation function mean through weight initialization. Ultimately, CFAT achieved an improvement of over 5% in accuracy.

**FIGURE 5.** Comparison display of each model in the CIC-IDS2017 dataset.

## F. SCALABILITY

According to Keuper et al. [55], deep neural networks have two specific bottlenecks: communication overhead and matrix operations. To address the issue of matrix operations, this paper eliminates scalability barriers by changing the method of computing attention weights using linear addition



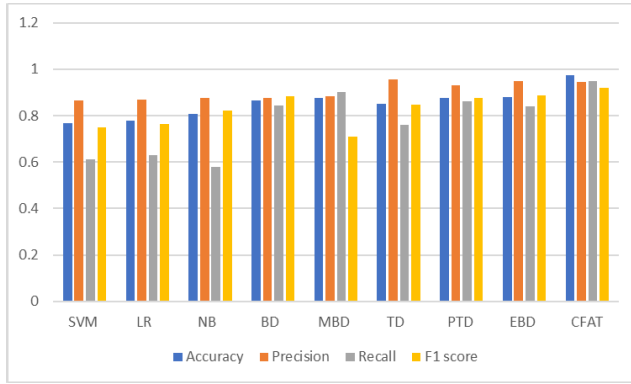


FIGURE 6. Comparison display of each model in the NSL-KDD dataset.

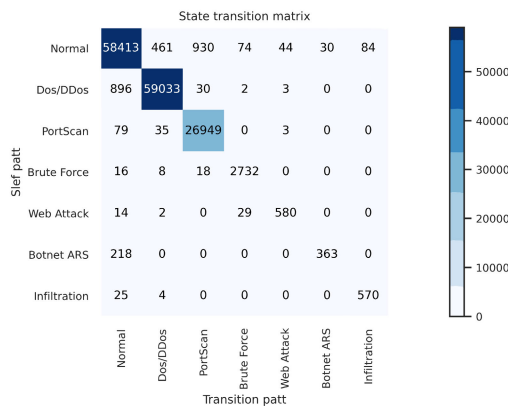


FIGURE 7. The confusion matrix of the CIC-IDS2017 dataset.

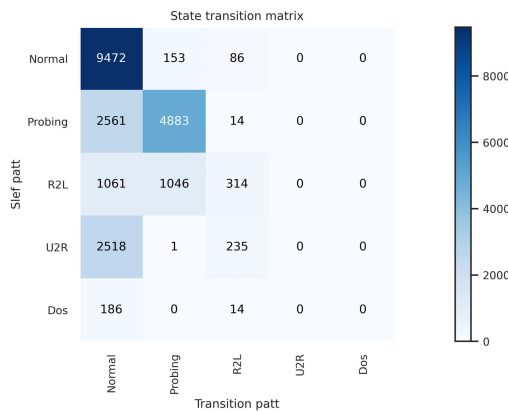


FIGURE 8. The confusion matrix of the NSL-KDD dataset.

instead of matrix operations. Regarding the problem of communication overhead, we deployed the model in the network security center, storing the training set on the local SSD of each working node, thus avoiding the overhead of accessing data during communication. Furthermore, considering the complexity of user operations and the dynamic nature of system state representation, which limits the scalability of deep models, this paper adopts a feature fusion method based on attention mechanisms to adapt to dynamic changes in real-world scenarios and allocate different weights to nodes in reality. There have been many studies on achieving scalability

through self-attention mechanisms, and these studies have been applied to practical scenarios by Hu et al. [56], Zhang et al. [57], and Zhou et al. [58].

## V. CONCLUSION

This paper introduces an improved BBO algorithm for feature selection in two datasets, employing the CFAT model for anomaly detection and prediction. Based on the utilization of Transformer's multi-head attention mechanism to acquire global feature information, CFAT redesigns a hierarchical connection structure to preserve both upper and lower-level feature information. Additionally, deep separable convolutions are employed to enhance the multi-head attention mechanism and further reduce computational complexity. Compared to the contrast experimental models used in this study, CFAT demonstrates significant improvements in all performance metrics, thereby better fulfilling the tasks of anomaly detection and prediction. However, it is noteworthy that both datasets exhibit severe imbalance, which is addressed in this study by utilizing OSS and Borderline-SMOTE methods. Although effective, there remains room for further optimization in future research. Future studies may explore binary text classification solutions for detecting malicious traffic and strategies for achieving real-time detection tasks.

## DATA AVAILABILITY

The data that support the findings of this study are openly available in Canadian Institute of Cybersecurity (CIC) Research Project at <http://205.174.165.80/CICDataset/CIC-IDS-2017/>.

## CONFLICTS OF INTEREST

The authors affirm that they have no conflicts of interest with respect to the publication of this paper.

## REFERENCES

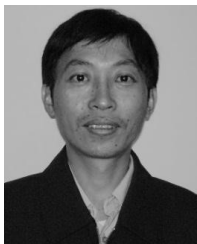
- [1] B. G. Goodarzi, H. Jazayeri, S. Fateri, and B. G. Goodarzi, "Intrusion detection system in computer network using hybrid algorithms (SVM and ABC)," *J. Adv. Comput. Res.*, vol. 5, no. 4, pp. 43–52, 2014.
- [2] D. S. Kim, H. N. Nguyen, S. Y. Ohn, and J. S. Park, "Fusions of GA and SVM for anomaly detection in intrusion detection system," in *Proc. Int. Symp. Neural Netw.*, 2005, pp. 415–420.
- [3] N. Ye, S. M. Emran, Q. Chen, and S. Vilbert, "Multivariate statistical analysis of audit trails for host-based intrusion detection," *IEEE Trans. Comput.*, vol. 51, no. 7, pp. 810–820, Jul. 2002.
- [4] M. T. Nguyen and K. Kim, "Genetic convolutional neural network for intrusion detection systems," *Future Gener. Comput. Syst.*, vol. 113, pp. 418–427, Dec. 2020.
- [5] Y. Duan, X. Li, X. Yang, and L. Yang, "Network security situation factor extraction based on random forest of information gain," in *Proc. 4th Int. Conf. Big Data Comput.*, 2019, pp. 194–197.
- [6] L. E. Jim and J. Chacko, "Decision tree based AIS strategy for intrusion detection in MANET," in *Proc. IEEE Region 10th Conf. (TENCON)*, Oct. 2019, pp. 1191–1195.
- [7] B. G. Narendrasinh and D. Vdevyas, "FLBS: Fuzzy lion Bayes system for intrusion detection in wireless communication network," *J. Central South Univ.*, vol. 26, no. 11, pp. 3017–3033, Nov. 2019.
- [8] H. Zhang, Y. Li, Z. Lv, A. K. Sangaiah, and T. Huang, "A real-time and ubiquitous network attack detection based on deep belief network and support vector machine," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 3, pp. 790–799, May 2020.

- [9] S. Longari, D. H. N. Valcarcel, M. Zago, M. Carminati, and S. Zanero, "CANnlo: An anomaly detection system based on LSTM autoencoders for controller area network," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 2, pp. 1913–1924, Jun. 2021.
- [10] G. C. Fernández and S. Xu, "A case study on using deep learning for network intrusion detection," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2019, pp. 1–6.
- [11] M. Mafarja, A. A. Heidari, M. Habib, H. Faris, T. Thaher, and I. Aljarah, "Augmented whale feature selection for IoT attacks: Structure, analysis and applications," *Future Gener. Comput. Syst.*, vol. 112, pp. 18–40, Nov. 2020.
- [12] T. A. Alamiedy, M. Anbar, Z. N. M. Alqattan, and Q. M. Alzubi, "Anomaly-based intrusion detection system using multi-objective grey wolf optimisation algorithm," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 9, pp. 3735–3756, Sep. 2020.
- [13] L. Liu, B. Xu, X. Zhang, and X. Wu, "An intrusion detection method for Internet of Things based on suppressed fuzzy clustering," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, pp. 1–7, Dec. 2018.
- [14] W. L. Al-Yaseen, A. K. Idrees, and F. H. Almasoudy, "Wrapper feature selection method based differential evolution and extreme learning machine for intrusion detection system," *Pattern Recognit.*, vol. 132, Dec. 2022, Art. no. 108912.
- [15] Z. Halim, M. N. Yousaf, M. Waqas, M. Sulaiman, G. Abbas, M. Hussain, I. Ahmad, and M. Hanif, "An effective genetic algorithm-based feature selection method for intrusion detection systems," *Comput. Secur.*, vol. 110, Nov. 2021, Art. no. 102448.
- [16] D. Simon, "Biogeography-based optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, Dec. 2008.
- [17] W. Guendouzi and A. Boukra, "GAB-BBO: Adaptive biogeography based feature selection approach for intrusion detection," *Int. J. Comput. Intell. Syst.*, vol. 10, no. 1, pp. 914–935, 2017.
- [18] D. Papamartzivanos, F. G. Mármol, and G. Kambourakis, "Dendron: Genetic trees driven rule induction for network intrusion detection systems," *Future Gener. Comput. Syst.*, vol. 79, pp. 558–574, Feb. 2018.
- [19] S. Roshan, Y. Miche, A. Akusok, and A. Lendasse, "Adaptive and online network intrusion detection system using clustering and extreme learning machines," *J. Franklin Inst.*, vol. 355, no. 4, pp. 1752–1779, Mar. 2018.
- [20] T. Hamed, R. Dara, and S. C. Kremer, "Network intrusion detection system based on recursive feature addition and bigram technique," *Comput. Secur.*, vol. 73, pp. 137–155, Mar. 2018.
- [21] S. Bhattacharya, S. R. S. Krishnan, P. K. R. Maddikunta, R. Kaluri, S. Singh, T. R. Gadekallu, M. Alazab, and U. Tariq, "A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU," *Electronics*, vol. 9, no. 2, p. 219, Jan. 2020.
- [22] Y. Chang, W. Li, and Z. Yang, "Network intrusion detection based on random forest and support vector machine," in *Proc. IEEE Int. Conf. Comput. Sci. Eng. (CSE) IEEE Int. Conf. Embedded Ubiquitous Comput. (EUC)*, vol. 1, Jul. 2017, pp. 635–638.
- [23] H. Ding, L. Chen, L. Dong, Z. Fu, and X. Cui, "Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection," *Future Gener. Comput. Syst.*, vol. 131, pp. 240–254, Jun. 2022.
- [24] X. Kan, Y. Fan, Z. Fang, L. Cao, N. N. Xiong, D. Yang, and X. Li, "A novel IoT network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network," *Inf. Sci.*, vol. 568, pp. 147–162, Aug. 2021.
- [25] C. Xu, J. Shen, and X. Du, "A method of few-shot network intrusion detection based on meta-learning framework," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3540–3552, 2020.
- [26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017. [Online]. Available: <https://arxiv.org/abs/1706.03762>
- [27] Y. Li, X. Peng, J. Zhang, Z. Li, and M. Wen, "DCT-GAN: Dilated convolutional transformer-based GAN for time series anomaly detection," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3632–3644, Apr. 2023.
- [28] Z. Chen, D. Chen, X. Zhang, Z. Yuan, and X. Cheng, "Learning graph structures with transformer for multivariate time-series anomaly detection in IoT," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9179–9189, Jul. 2021.
- [29] S. Tuli, G. Casale, and N. R. Jennings, "TranAD: Deep transformer networks for anomaly detection in multivariate time series data," in *Proc. VLDB Endowment*, vol. 15, 2022, pp. 1201–1214.
- [30] X. Han, S. Cui, S. Liu, C. Zhang, B. Jiang, and Z. Lu, "Network intrusion detection based on n-gram frequency and time-aware transformer," *Comput. Secur.*, vol. 128, May 2023, Art. no. 103171.
- [31] J. Guo, K. Han, H. Wu, Y. Tang, X. Chen, Y. Wang, and C. Xu, "CMT: Convolutional neural networks meet vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12165–12175.
- [32] C. Wu, F. Wu, T. Qi, Y. Huang, and X. Xie, "Fastformer: Additive attention can be all you need," 2021, *arXiv:2108.09084*.
- [33] I. Ullah and Q. H. Mahmoud, "Design and development of a deep learning-based model for anomaly detection in IoT networks," *IEEE Access*, vol. 9, pp. 103906–103926, 2021.
- [34] A. A. U. Rakhmonov, B. Subramanian, B. Olimov, and J. Kim, "Extensive knowledge distillation model: An end-to-end effective anomaly detection model for real-time industrial applications," *IEEE Access*, vol. 11, pp. 69750–69761, 2023.
- [35] A. Kim, M. Park, and D. H. Lee, "AI-IDS: Application of deep learning to real-time Web intrusion detection," *IEEE Access*, vol. 8, pp. 70245–70261, 2020.
- [36] F. Alin, A. Chemchem, F. Nolot, O. Flauzac, and M. Krajecki, "Towards a hierarchical deep learning approach for intrusion detection," in *Proc. Int. Conf. Mach. Learn. Netw.*, 2019, pp. 15–27.
- [37] A. Henry, S. Gautam, S. Khanna, K. Rabie, T. Shongwe, P. Bhattacharya, B. Sharma, and S. Chowdhury, "Composition of hybrid deep learning model and feature optimization for intrusion detection system," *Sensors*, vol. 23, no. 2, p. 890, Jan. 2023.
- [38] H. C. Altunay and Z. Albayrak, "A hybrid CNN+ LSTM-based intrusion detection system for industrial IoT networks," *Eng. Sci. Technol., Int. J.*, vol. 38, Feb. 2023, Art. no. 101322.
- [39] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Anomaly based network intrusion detection for IoT attacks using deep learning technique," *Comput. Electr. Eng.*, vol. 107, Apr. 2023, Art. no. 108626.
- [40] B. Olimov, S. Karshiev, E. Jang, S. Din, A. Paul, and J. Kim, "Weight initialization based-rectified linear unit activation function to improve the performance of a convolutional neural network model," *Concurrency Comput., Pract. Exper.*, vol. 33, no. 22, Nov. 2021, doi: [10.1002/cpe.6143](https://doi.org/10.1002/cpe.6143).
- [41] D. Z. Rodríguez, O. D. Okey, S. S. Maidin, E. U. Udo, and J. H. Kleinschmidt, "Attentive transformer deep learning algorithm for intrusion detection on IoT systems using automatic explainable feature selection," *PLoS One*, vol. 18, no. 10, Oct. 2023, Art. no. e0286652.
- [42] R. Yao, N. Wang, P. Chen, D. Ma, and X. Sheng, "A CNN-transformer hybrid approach for an intrusion detection system in advanced metering infrastructure," *Multimedia Tools Appl.*, vol. 82, no. 13, pp. 19463–19486, May 2023.
- [43] S. S. Kareem, R. R. Mostafa, F. A. Hashim, and H. M. El-Bakry, "An effective feature selection model using hybrid metaheuristic algorithms for IoT intrusion detection," *Sensors*, vol. 22, no. 4, p. 1396, Feb. 2022.
- [44] M. Jeyaselvi, R. K. Dhanaraj, M. Sathya, F. H. Memon, L. Krishnasamy, K. Dev, W. Ziyue, and N. M. F. Qureshi, "A highly secured intrusion detection system for IoT using EXPSO-STFA feature selection for LAANN to detect attacks," *Cluster Comput.*, vol. 26, no. 1, pp. 559–574, Feb. 2023.
- [45] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in *Proc. Int. Conf. Signal Process. Commun. Eng. Syst.*, Jan. 2015, pp. 92–96.
- [46] S. T. Mehedi, A. Anwar, Z. Rahman, and K. Ahmed, "Deep transfer learning based intrusion detection system for electric vehicular networks," *Sensors*, vol. 21, no. 14, p. 4736, Jul. 2021.
- [47] O. Elnakib, E. Shaaban, M. Mahmoud, and K. Emara, "EIDM: Deep learning model for IoT intrusion detection systems," *J. Supercomput.*, vol. 79, no. 12, pp. 13241–13261, Aug. 2023.
- [48] P. Choobdar, M. Naderan, and M. Naderan, "Detection and multi-class classification of intrusion in software defined networks using stacked auto-encoders and CICIDS2017 dataset," *Wireless Pers. Commun.*, vol. 123, no. 1, pp. 437–471, Mar. 2022.
- [49] A. S. Jaradat, M. M. Barhoush, and R. B. Easa, "Network intrusion detection system: Machine learning approach," *Indonesian J. Elect. Eng. Comput. Sci.*, vol. 25, no. 2, pp. 1151–1158, Feb. 2022.
- [50] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

- [51] S. Gamage and J. Samarabandu, "Deep learning methods in network intrusion detection: A survey and an objective comparison," *J. Netw. Comput. Appl.*, vol. 169, Nov. 2020, Art. no. 102767.
- [52] M. Alazab, R. A. Khurma, A. Awajan, and D. Camacho, "A new intrusion detection system based on Moth-Flame optimizer algorithm," *Expert Syst. Appl.*, vol. 210, Dec. 2022, Art. no. 118439.
- [53] J. Gu, L. Wang, H. Wang, and S. Wang, "A novel approach to intrusion detection using SVM ensemble with feature augmentation," *Comput. Secur.*, vol. 86, pp. 53–62, Sep. 2019.
- [54] B. A. U. Olimov, K. C. Veluvolu, A. Paul, and J. Kim, "UzADL: Anomaly detection and localization using graph Laplacian matrix-based unsupervised learning method," *Comput. Ind. Eng.*, vol. 171, Sep. 2022, Art. no. 108313.
- [55] J. Keuper and F.-J. Preundt, "Distributed training of deep neural networks: Theoretical and practical limits of parallel scalability," in *Proc. 2nd Workshop Mach. Learn. HPC Environments (MLHPC)*, Nov. 2016, pp. 19–26.
- [56] S. Hu, F. Zhu, X. Chang, and X. Liang, "UPDeT: Universal multi-agent reinforcement learning via policy decoupling with transformers," 2021, *arXiv:2101.08001*.
- [57] T. Zhang, H. Xu, X. Wang, Y. Wu, K. Keutzer, J. E. Gonzalez, and Y. Tian, "Multi-agent collaboration via reward attribution decomposition," 2020, *arXiv:2010.08531*.
- [58] T. Zhou, F. Zhang, K. Shao, K. Li, W. Huang, J. Luo, W. Wang, Y. Yang, H. Mao, B. Wang, D. Li, W. Liu, and J. Hao, "Cooperative multi-agent transfer learning with level-adaptive credit assignment," 2021, *arXiv:2106.00517*.



**XIAOBO FU** is currently pursuing the master's degree with the School of Computer Information, China Three Gorges University. His research interests include deep learning and computer science.



**TINGYAO JIANG** was born in 1969. He received the master's and Ph.D. degrees from the School of Computer Science and Technology, Huazhong University of Science and Technology, in 2001 and 2004, respectively. He is currently a Professor and the Ph.D. Supervisor with the College of Computer and Information Technology, China Three Gorges University. His research interests include artificial intelligence and information security.



**MIN WANG** was born in 1983. She received the master's degree from the College of Computer and Information Technology, China Three Gorges University, in 2009. She is currently a Lecturer with the School of Electronic Information, Hubei Three Gorges Polytechnic. Her research interests include machine learning and information security.

...