IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 February 2024, accepted 19 February 2024, date of publication 21 February 2024, date of current version 28 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3368392

RESEARCH ARTICLE

Intrusion Detection Model for Internet of Vehicles Using GRIPCA and OWELM

KAIJUN ZHANG[1], JIAYU YANG[2], YANGFEI SHAO[3], LEHUA HU[4], WEI OU[1], WENBAO HAN[1], AND QIONGLU ZHANG[5]

[1]School of Cyberspace Security (School of Cryptology), Hainan University, Haikou, Hainan 570228, China
[2]School of Computer Science and Technology, Hainan University, Haikou, Hainan 570228, China
[3]School of Information and Communication Engineering, Hainan University, Haikou, Hainan 570228, China
[4]School of Information Engineering, Hunan University of Science and Engineering, Yongzhou, Hunan 425199, China
[5]State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

Corresponding author: Lehua Hu (lehuahu2023@163.com)

This work was supported in part by the Joint Funds of National Natural Science Foundation of China under Grant U23A20304, in part by the Hainan Provincial Natural Science Foundation of China under Grant 621RC508, in part by the State Key Laboratory of Information Security under Grant 2022-MS-04, in part by the Henan Key Laboratory of Network Cryptography Technology under Grant LNCT2021-A16, in part by the Science Project of Hainan University under Grant KYQD(ZR)-21075, and in part by the High Tech Industry and Technology Innovation Leadership Project in Hunan Province under Grant 2020SK2022.

ABSTRACT With the rapid development of the Internet of Vehicles, a large amount of vehicle network data is being generated. The large amount of data presents network communication security challenges. Although intrusion detection technology can assist in safeguarding the system from malicious attacks, the substantial data generated within the vehicle network poses time-consuming detection challenges. Thus, we propose an intrusion detection model for the Internet of Vehicles, utilizing Gaussian random incremental principal component analysis (GRIPCA) and optimal weighted extreme learning machine (OWELM). First, we utilize GRIPCA to reduce data redundancy by projecting high-dimensional data into a low-dimensional space, thus reducing storage costs. Then, we utilize the dynamic inertia weight particle swarm optimization (DPSO) to optimize the parameters of the weighted extreme learning machine (WELM) to achieve the best performance. We utilize the NSL-KDD and CIC-IDS-2017 datasets to perform experiments and compare the results with other techniques. The experimental results show the excellence of the proposed model, achieving an accuracy rate of 91.02% on the NSL KDD dataset and 94.67% on the CIC-IDS-2017 dataset.

INDEX TERMS Extreme learning machine, Internet of Vehicles, intrusion detection, particle swarm optimization, principal component analysis.

I. INTRODUCTION

In recent years, the Internet of Vehicles has emerged as a new paradigm for intelligent transport systems, aiming to enhance road safety and driving efficiency [1]. The Internet of Vehicles constitutes an open, integrated network system established through the interconnection of vehicles, wireless networks, and other units [2]. In the fields of intelligent transportation and autonomous driving, the importance of the Internet of Vehicles is evident as a crucial element of the intelligent city network [3]. The Internet of Vehicles technology facilitates the cooperation of diverse vehicles through intelligent planning and scheduling, leading to smoother traffic flow, real-time information exchange, vehicle interaction, and mitigation of traffic accidents caused by human factors. Vehicles can also enhance their intelligence by establishing extensive external connectivity [4]. Intelligent vehicles fulfill practical needs, including driving safety reminders, video surveillance, and real-time weather forecasts, thus enhancing the overall service experience for travelers. The Internet of Vehicles has enormous potential to contribute to traffic management, environmental monitoring, and public safety.

With the number of interconnected vehicles increasing, they face significant security issues [5]. The security issues are as follows. (1) Data privacy leakage: Vehicles collect and transmit data about their location and drivers. Inadequate

The associate editor coordinating the review of this manuscript and approving it for publication was Tony Thomas.

data protection measures can result in data leakage, posing a significant threat to the privacy of both vehicle owners and passengers. In order to deal with the risk of data privacy leakage, robust encryption algorithms can be used to protect data transmitted on the Internet of Vehicles to prevent unauthorized access and interception. When sharing data, essential information such as the identity and location of individual vehicles is anonymized to reduce the risk of privacy leaks. (2) Hacker attack: Hackers engage in unauthorized malicious activities targeting vehicles. Attackers exploit vulnerabilities in vehicle systems to gain unauthorized access and potentially disrupt the Internet of Vehicles system. In order to deal with hacker attacks, encryption technology can be used to protect communications between vehicles and prevent information leakage and tampering. Implement multi-level authentication to ensure only authorized users have access to vehicle systems. (3) Network communication security: Vehicle connectivity relies on network communication, rendering vehicle systems vulnerable to network security threats. In order to deal with network communication security risks, real-time monitoring systems can be deployed to monitor network traffic and activities in the Internet of Vehicles in real-time. Promptly detect and respond to potential intrusions to reduce the impact of attacks. Applying machine learning algorithms enables intrusion detection models to learn normal behaviors in the Internet of Vehicles and identify abnormal patterns. Machine learning can help systems detect new, unknown attacks. (4) External link device security risks: As the functions carried by intelligent connected vehicles gradually increase, the frequent access of external ecological components to the vehicle will bring new security risks. When consumers purchase and install external link products for vehicles, they will bring the risk of external virus intrusion attacks. In order to address external link device security risks, the software and firmware of external devices can be updated and patched promptly to eliminate known vulnerabilities. Develop a reasonable update strategy to ensure devices can promptly obtain and install the latest security patches. Isolate external devices and use virtualization or container technology to reduce the impact of external devices on the entire vehicle system.

The rising number of cyber-attacks on the Internet of Vehicles has prompted concerns about the stability of the Internet of Vehicles, which could result in serious consequences such as unavailable vehicles or traffic accidents. To address this issue, intrusion detection models have become a key factor in ensuring the security of Internet of Vehicles network communication. Advanced security technologies are used to monitor and detect potential intrusion activities, aiming to maintain the safety of both vehicles and passengers. Intrusion detection methods can identify signs of network attacks within the Internet of Vehicles by collecting and analyzing communication flow data, achieving the classification of intrusion and normal patterns. This security measure contributes to maintaining the stability and safety of the Internet of Vehicles.

Intrusion detection has received more and more research attention in ensuring the security of network communication in the Internet of vehicles [6]. In the Internet of Vehicles context, limited computing capacity poses a significant challenge to reducing the complexity of extensive data in vehicle networks [7]. Additionally, the high-speed movement of vehicles requires minimizing the detection time. Therefore, we must design a lightweight intrusion detection model for the Internet of Vehicles. We make the following contributions.

- Propose the GRIPCA to reduce the required storage space and computation time. We project the vehicle network data using a Gaussian random matrix, and then the data is processed in batches to decrease the dimensionality of the vehicle network data.
- Propose the OWELM to enhance the performance of intrusion detection. We introduce a weight matrix to assign variable weights to different samples and employ a dynamic inertial weight particle swarm technique to optimize the parameters of the algorithm.
- Evaluate the performance of existing techniques using the NSL-KDD and CIC-IDS-2017 datasets to authenticate the effectiveness of our proposed method.

The rest of the paper is organized as follows. Section II details an overview of the relevant research about intrusion detection in the Internet of Vehicles. Section III details the intrusion detection model. Section IV overviews our experimental environment, datasets, evaluation metrics, and analysis results. Section V provides a summary of the paper.

II. RELATED WORKS

Intrusion detection methods can be grouped into three categories: traditional machine learning algorithms, deep learning algorithms, and other types (non-machine learning algorithms).

Intrusion detection methods based on traditional machine learning algorithms:

Lu [8] designed an energy-aware intrusion detection model to manage the secure data transmission method of the Internet of Vehicles and improve the accuracy and precision of detecting existing attacks. Aliyu et al. [9] explored a statistical adversarial detector to identify unrecognized adversarial samples. Anyanwu et al. [10] implemented a lightweight False BSM Detection Scheme, capitalizing on hyper-parameter tuning with the ensemble random forest classifier to precisely categorize attack types in the Internet of Vehicles. Alsarhan et al. [11] applied support vector machines for intrusion detection, and the results showed that GA was superior to other optimization algorithms. Rani and Sharma [12] proposed an intelligent transportation system based on the Internet of Vehicles in smart city scenarios. The results show that the proposed system can provide high detection accuracy and low computational cost through ensemble learning and average important feature selection. Li et al. [13] proposed an intrusion detection method for Internet of Vehicles based on transmission double-depth Q network. Wang et al. [14] designed a lightweight intrusion

detection method that employs MobileNetv2 as the backbone, integrating transfer learning techniques and hyper-parameter optimization methods.

Intrusion detection methods based on deep learning algorithms: Nie et al. [15] introduced a deep learning architecture that relies on convolutional neural networks to extract features from link loads and detect intrusions targeting vehicles. Yang et al. [16] developed an Internet of Vehicles intrusion detection model based on ConvLSTM and used ConvLSTM to significantly reduce the model size and convergence time. Ahmed et al. [17] designed a deep learning-based intrusion detection system to protect vehicle data from malicious attacks. Grover et al. [18] introduced a multi-layer heterogeneous vehicle network based on edge computing for Internet of Vehicles services. Additionally, they introduced an unsupervised vehicle behavior detection algorithm based on stacked long short-term memory to enhance communication security among participating vehicles. Hu et al. [19] designed a two-dimensional mosaic pattern encoding convolutional neural network anomaly detection model. The model fully leveraged the capabilities of convolutional neural networks to extract grid data. Yang et al. [20] designed a detection model for the Internet of Vehicles that relies on transfer learning and ensemble learning techniques, incorporating convolutional neural networks and hyperparameter optimization for improved performance. Alferaidi et al. [21] introduced an intrusion detection method that utilizes the Apache Spark framework. The framework captured features for detecting network intrusions in vehicle networks and identifying abnormal behavior. Xing et al. [22] proposed a technique involving the parallel analysis of spatiotemporal features for intrusion detection in vehicle networks. The method significantly improved the performance.

Intrusion detection methods based on other types (non-machine learning algorithms):

Panda et al. [23] proposed a new decision support framework for Internet of Vehicles honeypot deception, using decoy systems such as honeypots to truly collect attacker data. Haydari et al. [24] proposed a statistical non-parametric intrusion detection system for the online detection of attacks, which has superior performance in fast and accurate detection. Zhao et al. [25] designed an intrusion detection system based on an electronic control unit clock skew. This system could detect spoofing attacks, bus-off attacks, and masquerade attacks. Zhang et al. [26] designed an improved intrusion detection algorithm that uses many-objective optimization to optimize the parameters. Yu et al. [27] introduced an intrusion detection method that relies on CAN message ID features, enhancing the detection accuracy of DoS and spoofing attacks.

As mentioned above, although numerous intrusion detection methods have been proposed to enhance Internet of Vehicles security, they predominantly focus on implementing the learning model algorithms and overlook the detection time [28]. Although most of the proposed schemes have achieved high attack detection accuracy, there is still room for improvement. The major disadvantages of existing models include high consumption of computational and storage resources, long training time, and manual selection of network configuration parameters. Therefore, we propose an intrusion detection model for Internet of Vehicles using GRIPCA and OWELM. We employ the feature dimensionality reduction technique of GRIPCA to reduce the data dimensionality, enhancing the efficiency of the intrusion detection model. Additionally, we utilize the DPSO to optimize the parameters of the WELM, constructing an efficient intrusion detection model.

III. INTRUSION DETECTION MODEL OF THE PROPOSED HPPELM

In this section, we present an intrusion detection model for the Internet of Vehicles. Our proposed model is divided into three parts. In the first part, we preprocess the vehicle network data. In the second part, we use the GRIPCA to reduce the dimensionality of the data. In the third part, we use the OWELM for intrusion detection. The flow of our model is shown in Figure 1. For ease of reference, the key notations of our model are shown in Table 1.

TABLE 1. Common key symbol.

Symbol	Meaning
v	Velocity of particles
w	Index of inertia
R	Gaussian random matrix
c	Particle learning factor
W	Input weight value
b	Hidden layer threshold
H	Hidden layer output matrix
β	Weight of output

A. DATA PREPROCESSING

We preprocess vehicle network data by performing missing value imputations, transformations, and standardization. Firstly, the vehicle network data is structured into a matrix where each row represents a data point, and each column represents a specific feature. Subsequently, we address missing values in this matrix by filling them with the mean value of the respective feature. We employ label encoding to convert character features into numerical features when dealing with character features within the data. The processed matrix is given in Equation (1).

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix}_{n\times m} \tag{1}$$

where n is the number of vehicle network data, m is the number of features in the vehicle network data, and $a_{n,m}$ denotes the value in the vehicle network data.

We standardize the vehicle network data to reduce the impact of magnitude variations among features on intrusion

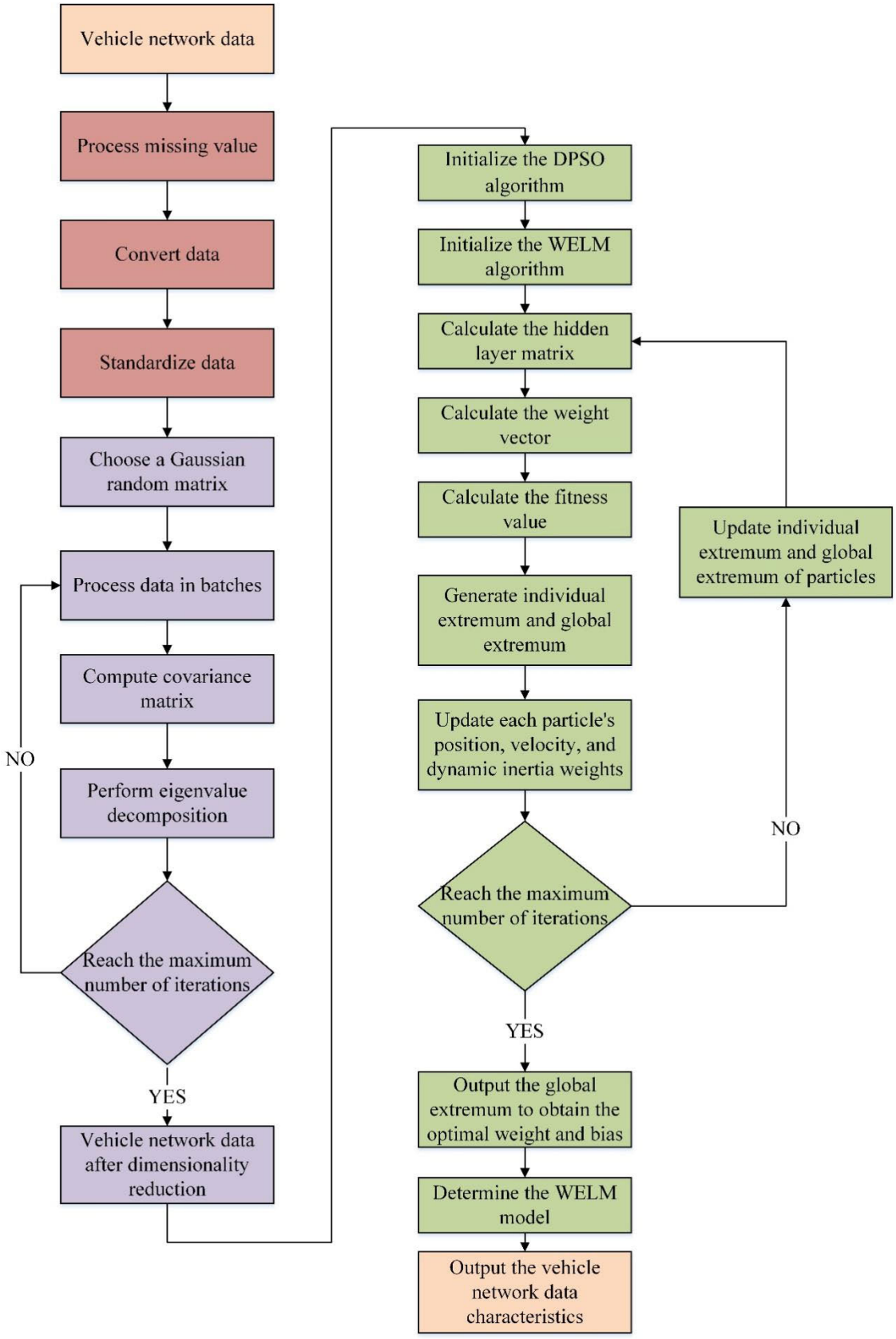


FIGURE 1. Intrusion detection model of Proposed HPPELM.

detection. We standardize data to ensure data consistency, comparability, and effectiveness of model training. Standardization eliminates scale differences between different sources and types of data, allowing models to learn features better, achieve accurate detection, improve algorithm performance, and enable systems to respond to potential security threats more effectively. Standardization is adjusting the data distribution to a standard normal distribution with a mean of 0 and a standard deviation of 1. The calculation formulas are shown in Equation (2)-(4).

$$\bar{a} = \frac{1}{n}\sum_{i=1}^{n} a_i \tag{2}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i - \bar{a})^2} \tag{3}$$

$$\hat{a}_{ij} = \frac{a_{ij} - \bar{a}}{\sigma} \tag{4}$$

where $\bar{a}$ is the mean of the vehicle network data and σ is the standard deviation of the vehicle network data.

The representation of the matrix after we standardize the vehicle network data is shown in Equation (5).

$$Y = \begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nm} \end{pmatrix}_{n\times m} \tag{5}$$

B. DATA DIMENSIONALITY REDUCTION BASED ON GRIPCA

We employ the GRIPCA for dimensionality reduction of vehicle network data. The approach reduces data redundancy by projecting high-dimensional data into a lower-dimensional space, reducing storage costs. The steps of data dimensionality reduction are as follows.

Step 1: We utilize a Gaussian random matrix to project the vehicle network data into a low-dimensional subspace. This random projection technique preserves the main structure and characteristics of the data. Firstly, we generate a Gaussian random matrix consisting of random variables obeying a Gaussian distribution. Next, we multiply the preprocessed vehicle network data with a Gaussian random matrix to obtain a new matrix.

Step 2: We divide the projected data into b batches and process them batch by batch. We subtract the mean of each feature from each batch of data and then calculate the covariance matrix of each batch of data. The covariance matrix provides the degree of correlation between various features in the data. The calculation formula is given in Equation (6).

$$C_i = \frac{1}{n-1} L_i^T L_i \tag{6}$$

where L_i^T represents the transpose of L_i.

Step 3: We perform eigenvalue decomposition on each covariance matrix to obtain eigenvalues and eigenvectors. These eigenvectors represent the principal components. The eigenvectors of each batch of data are combined with the eigenvectors of the previous batch to update the principal components. We can update it by the rule of accumulates feature values.

Step 4: After processing all data, we calculate the variance contribution rate of each principal component. The variance contribution rate is the ratio of each feature value divided by the total variance. The cumulative contribution rate is the sum of the variance contribution rates of the first p principal components. The calculation formula of the cumulative contribution rate is given in Equation (7).

$$\varphi_p = \frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{k} \lambda_i} \tag{7}$$

where k is the number of total eigenvalues, and p is the number of selected principal components.

Step 5: We select the number of principal components to retain through the target of the cumulative contribution rate. We use experiments and empirical judgments from other papers to select a cumulative contribution rate threshold to meet actual needs, ensure that the selected principal components can explain most of the variance, and simplify the data and model. Such selection criteria can balance data retention and dimensionality reduction, allowing principal component analysis to be better applied to data analysis and interpretation. We use the selected number of principal components to construct a projection matrix. Finally, the data are projected onto the selected principal components to obtain dimensionally reduced vehicle network data.

The pseudo-code of data dimensionality reduction is shown in Algorithm 1.

Algorithm 1 Data Dimensionality Reduction Based on GRIPCA

Input: X: Vehicle network data; R: Random matrix;
Output: The matrix after dimensionality reduction S;
1: Centralize data matrix X;
2: Replace missing values in the data with average values;
3: Perform normalization to obtain the matrix Y;
4: Pick a Gaussian random matrix R;
5: $L = YR$ // Obtain network data after projection dimension reduction;
6: **While** the Unreached batch **do**
7: Calculate the mean for each small batch of data and accumulate the population mean;
8: Subtract the mean value of each small batch of data;
9: Calculate the covariance matrix $C_i = \frac{1}{n-1} L_i^T L_i$;
10: Update the principal component;
11: **end while**
12: Project the data to the selected principal components;

C. INTRUSION DETECTION BASED ON OWELM

We use the DPSO to optimize the parameters of the WELM for intrusion detection. The WELM is an improvement

upon ELM, offering advantages such as rapid training, simplicity, suitability for high-dimensional data, and good generalization capabilities. By introducing a weight matrix, WELM exhibits increased flexibility in handling relationships between features, enhancing the model's ability to generalize. This is particularly beneficial when dealing with high-dimensional data and scenarios demanding real-time responsiveness. Furthermore, the DPSO incorporates the cosine function from trigonometry. Periodic oscillations are characteristic of cosine functions. The DPSO significantly enhances global search capability and adaptability by introducing a dynamic inertia weight strategy. The particles flexibly explore the search space during the search process, enhancing convergence speed and efficiency. The steps for intrusion detection are as follows.

Step 1: We initialize the DPSO by randomly generating particles. Each particle represents a combination of weights and biases from the WELM. These particles have positions corresponding to the values of the weights and biases. The speed of a particle represents the direction and distance the particle moves in the search space. We establish upper and lower bounds for each weight and bias in the WELM to prevent the search space from becoming too broad or overly constrained. We use the initialization of the learning factor to adjust the speed and direction of the particle movement in the search space. We initialize the particle velocity to keep the particle velocity within the specified range. It prevents particles from moving too fast in the search space.

Step 2: We use negative accuracy as the fitness function, choose an appropriate activation function for each neuron, and randomly initialize the weights connected to the input features and bias terms. We start the iterative optimization process. Then, we train WELM with existing parameters and evaluate the output weight matrix and negative accuracy. The calculation formulas are shown in Equation (8)-(11).

$$H\beta = Y \tag{8}$$

$$\hat{\beta} = \begin{cases} H^T \left(I + WHH^T\right)^{-1} WY, N < L \\ \left(I + WHH^T\right)^{-1} H^T WY, N > L \end{cases} \tag{9}$$

$$W = \frac{1}{Count\ (t_i)} \tag{10}$$

$$f\ (x) = h\ (x)\ \hat{\beta} = \begin{cases} h(x)H^T \left(I + WHH^T\right)^{-1} WY, N < L \\ h\ (x) \left(I + WHH^T\right)^{-1} H^T WY, N > L \end{cases} \tag{11}$$

Among them, H is the output matrix of the hidden layer, β is the connection weight between the hidden layer and the output layer, $Count\ (t_i)$ is the number of samples of the category in the training sample, and $f\ (x)$ is the output function of WELM.

Step 3: We evaluate the individual and global optimal values of all particles in the DPSO, calculate the fitness value of each particle, and compare the current individual extreme value with the global extreme value to select the individual with smaller negative accuracy. The calculation formulas of individual extreme value and global extreme value are shown in Equation (12)-(13).

$$p_{ib} = \begin{cases} p_i \left(A_{P_i} < A_{P_{ib}}\right) \\ p_{ib} \left(A_{P_i} > A_{P_{ib}}\right) \end{cases} \tag{12}$$

$$p_g = \begin{cases} p_i \left(A_{P_i} < A_{P_g}\right) \\ p_g \left(A_{P_i} > A_{P_g}\right) \end{cases} \tag{13}$$

Among them, A_{P_i} is the accuracy of the i-th particle, $A_{P_{ib}}$ is the optimal accuracy of the i-th particle, and A_{P_g} is the optimal accuracy of all particles.

Step 4: We update the position, velocity, and dynamic inertia weights of the DPSO after each iteration. The formulas are shown in Equation (14)-(16). The process terminates when the maximum number of iterations is reached. Finally, we store the parameters of the optimal WELM in the global extremum.

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{14}$$

$$v_i^{t+1} = wv_i^t + c_1 r_1 \left(p_{ib}^t - x_i^t\right) + c_2 r_2 \left(p_g^t - x_i^t\right) \tag{15}$$

$$\mathrm{w} = w_{max} + (w_{min} - w_{max}) \times \frac{1 + \cos\left(\frac{t}{t_{max}} \times \pi\right)}{2} \tag{16}$$

where t is the current number of iterations, c_1 and c_2 are the learning factors of the particle, representing the local and global learning factors of the particle respectively, w is the inertia weight, r_1 and r_2 are random numbers between [0, and the range of the velocity is $[v_{min}, v_{max}]$.

Step 5: We provide the optimized parameters to the WELM. The WELM computes the output weights using these values, resulting in the OWELM. This model is utilized to identify normal and abnormal traffic in the vehicle network environment.

The pseudo-code of intrusion detection is shown in Algorithm 2.

IV. RESULTS AND DISCUSSION

A. ENVIRONMENT

The experimental environment of the intrusion detection model is shown in Table 2.

B. DATASET DESCRIPTION

There is no publicly available dataset for inter-vehicle communication due to the privacy or disinclination of business shareholders to share their data with academia. It is defensible yet to use a dataset based on general networks for intrusion detection model in the Internet of Vehicles [29]. We select the NSL-KDD dataset [30] and the CIC-IDS-2017 dataset [31] to test the intrusion detection performance of our proposed model. The NSL-KDD dataset is one of the most used datasets in the field of network intrusion detection. In the NSL-KDD dataset, each network traffic record consists of 42 features, which 38 dimensions are digital

Algorithm 2 Intrusion Detection Based on OWELM

Input: S:The processed data; w:Initialize the input weight; b:Initialize the hidden layer bias; T:Initialize the number of iterations; c:Initialize the learning factor;

Output: Vehicle network data characteristics;

1: Initialize the parameters of the WELM;

2: Determine the fitness function of the particle swarm;

3: **while** $T > t$ **do**

4: Train WELM and calculate the negative accuracy;

$$\hat{\beta} = \begin{cases} H^T \left(I + WHH^T\right)^{-1} WY, N < L \\ \left(I + WHH^T\right)^{-1} H^T WY, N > L \end{cases}$$ // Calculate the output weight;

$W = \frac{1}{Count(t_i)}$ // Calculate sample weight;

$f(x) = h(x)\hat{\beta}$ //Output function;

5: Calculate the extreme value of all particles in a particle swarm;

6: Update the position, velocity, and dynamic inertia weights of each particle;

$x_i^{t+1} = x_i^t + v_i^{t+1}$ //Update the position of each particle;

$v_i^{t+1} = wv_i^t + c_1 r_1 \left(p_{ib}^t - x_i^t\right) + c_2 r_2 \left(p_g^t - x_i^t\right)$ //Update the velocity of each particle;

$$\mathrm{w} = w_{max} + (w_{min} - w_{max}) \times \frac{1+\cos\left(\frac{t}{t_{max}} \times \pi\right)}{2}$$ //Update inertia weight;

7: Output optimal input weight and hidden layer bias;

8: **end while**

9: Train the OWELM;

TABLE 2. Configuration.

	Tool	Version
Hardware	CPU	Intel Core i7-12650H 12-core 2.3GHz
	RAM	16GB
	GPU	GeForce RTX 3050
Software	Operating system	Ubuntu 20.04 LTS
	Python	Python 3.8.15
	Anaconda	conda 22.9.0
	Pycharm	Pycharm 2020

TABLE 3. Data distribution of NSL-KDD dataset.

Class label	Number of instances		
	Train	Test	Total
Normal	67343	9711	77054
Dos	45927	7458	53385
Probe	11656	2421	14077
U2R	52	200	252
R2L	995	2754	3749
Total	125973	22544	148517

features, 3 dimensions are character features, and the remaining dimension is the label feature. The attack behaviors in the NSL-KDD dataset are primarily categorized into DoS, Probe, U2R, and R2L. The data distribution of the NSL-KDD dataset is shown in Table 3. The CIC-IDS-2017 dataset was released in 2017 and is widely used in intrusion detection experiments. It was captured over five days and includes over 2.5 million records and 78 features, including label columns. The attack labels in the CIC-IDS-2017 dataset include DoS attacks, injection attacks, brute force attacks, and web attacks. The data distribution of the CIC-IDS-2017 dataset is shown in Table 4. In the division of experimental datasets, the NSL-KDD dataset has been publicly processed, and its train set and test set have been divided. We divide the train set and test set of the CIC-IDS-2017 dataset into 80% and 20%.

TABLE 4. Data distribution of CIC-IDS-2017 dataset.

Class label	Number of instances		
	Train	Test	Total
BENIGN	1818097	455000	2273097
DOS	305047	75652	380699
PortScan	126988	31942	158930
Brute Force	11089	2746	13835
Web Attack	1778	402	2180
Bot	1568	398	1966
Infiltration	27	9	36
Total	2264594	566149	2830743

C. EVALUATION METRICS

We employ Accuracy, Precision, Recall, FPR, F1-score, and Detection time to evaluate the performance of our proposed method. Specifically, Accuracy is the proportion of correctly predicted samples among the total number of samples. Precision measures the percentage of predicted abnormal samples that are genuinely abnormal among all the samples predicted as abnormal. The recall is the proportion of samples correctly classified as abnormal when they are abnormal. FPR denotes the fraction of samples erroneously classified as normal among those abnormal samples. F1-score is calculated as the harmonic mean of Precision and Recall. Detection time is the duration required to identify all samples. The calculation formulas are shown in Equation (17)-(21).

$$Accurary = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

$$FPR = \frac{FP}{TN + FP} \tag{20}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{21}$$

where TP (True Positive) counts anomalous samples correctly classified as anomalies, TN (True Negative) counts normal samples correctly classified as normal, FP (False Positive) counts normal samples mistakenly classified as anomalies, and FN (False Negative) counts anomalous samples mistakenly classified as normal.

D. RESULTS AND ANALYSIS

1) EVALUATION ON THE NSL-KDD DATASET

According to the GRIPCA, we set the cumulative contribution rate of features to 95% [32]. The cumulative contribution rate of feature vectors after feature dimensionality reduction on the NSL-KDD dataset is shown in Table 5.

We can observe from Table 5 that after applying dimensionality reduction to the data, the cumulative contribution rate of the nine features reaches 96.44%, indicating that these nine features can represent 96.44% of the information within the entire dataset.

TABLE 5. Cumulative contribution rate of features on the NSL-KDD dataset.

Serial number	Cumulative contribution rate
1	41.44%
2	70.28%
3	77.01%
4	82.66%
5	87.49%
6	90.45%
7	93.02%
8	94.98%
9	96.44%

We utilize the DPSO to optimize the WELM. The parameters of DPSO are configured as follows: the population size of the particle swarm is 30, the iterations are 100, the learning factor is 2, the maximum particle position is 1, the minimum particle position is −1, the maximum particle velocity is 1, and the minimum particle velocity is −1. The changes of the fitness values of the DPSO on the NSL-KDD dataset are depicted in Figure 2. The table shows that the DPSO iterations reach 90 and the WELM reaches convergence and attains the maximum accuracy.

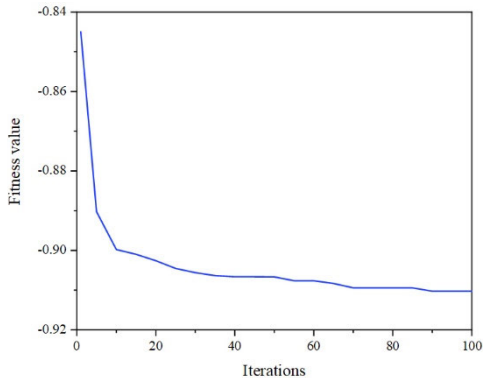


FIGURE 2. The changes of the fitness values of the DPSO algorithm on the NSL-KDD dataset.

We use the OWELM for detection and obtain experimental results. The experimental results on the NSL-KDD dataset are shown in Table 6.

To further verify the effectiveness of HPPELM, we compare it with CNN-IDS [15], LSTM-IDS [18], SPARK-IDS [21], GASVM-IDS [11], DDQN-IDS [13]. We conduct experiments in the same experimental environment. The comparative experiments are introduced as follows.

- CNN-IDS: In 2020, Nie L et al. developed an efficient data-driven Intrusion Detection System by examining the link load within the Internet of Vehicles. They introduced a CNN-based deep architecture and formulated a loss function to improve the convergence of training errors.
- LSTM-IDS: In 2021, Grover H et al. introduced a secure, efficient, and intelligent multi-layer heterogeneous Internet of Vehicles network. A long and short-term memory algorithm protected the communication between participating vehicles. The results indicate that the proposed model outperforms alternative approaches.
- SPARK-IDS: In 2022, Alferaidi A et al. introduced an intrusion detection method for Internet of Vehicles utilizing the Apache Spark framework. The method reduces detection time, enhances the detection rate, and satisfactorily meets real-time demands.
- GASVM-IDS: In 2023, Alsarhan A et al. applied support vector machines for intrusion detection and used heuristic algorithms to optimize the task. Support vector machines have high generalization performance, can handle small samples, and can effectively handle the curse of dimensionality.
- DDQN-IDS: In 2023, Li Z et al. proposed an intrusion detection method for the Internet of Vehicles based on the transmission double-depth Q network. Experimental results show that the proposed method improves F1-score and detection accuracy and reduces time consumption and convergence time.

We evaluate the proposed method by measuring the Accuracy, Precision, Recall, FPR, F1-score, and Detection time. The confusion matrix serves as a summary of the results of intrusion detection. It employs count values to summarize the normal traffic and abnormal traffic. The confusion matrix on the NSL-KDD dataset is shown in Figure 3.

Figure 4 shows the Accuracy of the proposed and existing methods like CNN-IDS, LSTM-IDS, SPARK-IDS, GASVM-IDS, and DDQN-IDS on the NSL-KDD dataset. Compared with other methods, the proposed HPPELM demonstrates superior accuracy for intrusion detection, achieving an accuracy rate of 91.02%. In contrast, CNN-IDS, LSTM-IDS, SPARK-IDS, GASVM-IDS, and DDQN-IDS achieve accuracy rates of 78.22%, 75.44%, 75.91%, 77.12%, and 84.78%, respectively.

Figure 5 shows the Precision of the proposed and existing methods on the NSL-KDD dataset. Compared with other methods, the proposed HPPELM demonstrates a high precision rate for intrusion detection, achieving a precision rate of 92.68%. In contrast, CNN-IDS, LSTM-IDS, SPARK-IDS, GASVM-IDS, and DDQN-IDS achieve precision rates of 87.86%, 92.51%, 87.61%, 96.67%, and 82.62%, respectively.

Figure 6 shows the Recall of the proposed and existing methods on the NSL-KDD dataset. Compared with other

TABLE 6. Classification results on the NSL-KDD dataset.

Model	Accuracy (%)	Precision (%)	Recall (%)	FPR (%)	F1-score (%)	Detection time (s)
HPPELM	91.02	92.68	91.45	9.5	92.06	0.005

methods, the proposed HPPELM demonstrates a high recall rate for intrusion detection, achieving a recall rate of 91.45%. In contrast, CNN-IDS, LSTM-IDS, SPARK-IDS, GASVM-IDS, and DDQN-IDS achieve recall rates of 71.63%, 61.86%, 67.19%, 61.73%, and 79.6%%, respectively.

Figure 7 shows the FPR of the proposed and existing methods on the NSL-KDD dataset. Compared with other methods, the proposed HPPELM demonstrates a notably lower false positive rate for intrusion detection, achieving a false positive rate of 9.5%. In contrast, CNN-IDS, LSTM-IDS, SPARK-IDS, GASVM-IDS, and DDQN-IDS achieve false positive rates of 13.07%, 6.61%, 12.56%, 2.5%, and 8.37%, respectively.

Figure 8 shows the F1-score of the proposed and existing methods on the NSL-KDD dataset. Compared with other methods, the proposed HPPELM demonstrates a high F1-score for intrusion detection, achieving an F1-score of 92.06%. In contrast, CNN-IDS, LSTM-IDS, SPARK-IDS, GASVM-IDS, and DDQN-IDS achieve the F1-score of 78.92%, 74.14%, 76.05%, 75.44%, and 85.62%, respectively.

Figure 9 shows the Detection time of the proposed and existing methods on the NSL-KDD dataset. The proposed HPPELM requires a shorter intrusion detection time than other methods. The detection time is 0.005 seconds. In contrast, the detection time for CNN-IDS, LSTM-IDS, SPARK-IDS, GASVM-IDS, and DDQN-IDS are 0.48 seconds, 2.11 seconds, 1.69 seconds, 15.44 seconds, and 12.67 seconds, respectively.

2) EVALUATION ON THE CIC-IDS-2017 DATASET

We perform the same experiments on the CIC-IDS-2017 dataset. The cumulative contribution rate of feature vectors after feature dimensionality reduction on the CIC-IDS-2017 dataset is shown in Table 7.

TABLE 7. The changes of the fitness values of the DPSO algorithm on the CIC-IDS-2017 dataset.

Serial number	Cumulative contribution rate
1	34.56%
2	62.31%
3	74.21%
4	81.84%
5	87.71%
6	90.34%
7	92.77%
8	94.72%
9	96.24%

We can observe from Table 7 that after applying dimensionality reduction to the data, the cumulative contribution rate of the nine features reaches 96.24%.

The changes in fitness values of the DPSO on the CIC-IDS-2017 dataset are depicted in Figure 10. The table shows that the DPSO iterations reach 25 and the WELM reaches convergence and attains the maximum accuracy.

We use the OWELM for detection and obtain experimental results. The experimental results on the CIC-IDS-2017 dataset are shown in Table 8.

We compare with other methods on the CIC-IDS-2017 dataset. The confusion matrix on the CIC-IDS-2017 dataset is shown in Figure 11.

The Accuracy of proposed and existing methods like CNN-IDS, LSTM-IDS, SPARK-IDS, GASVM-IDS, and DDQN-IDS on the CIC-IDS-2017 dataset is shown in Figure 12. Compared with other methods, the proposed HPPELM demonstrates superior accuracy for intrusion detection, achieving an accuracy rate of 94.67%. In contrast, CNN-IDS, LSTM-IDS, SPARK-IDS, GASVM-IDS, and DDQN-IDS achieve accuracy rates of 84.67%, 86.38%, 93.66%, 91.77%, and 92.61%, respectively.

The Precision of proposed and existing methods on the CIC-IDS-2017 dataset is shown in Figure 13. Compared with other methods, the proposed HPPELM demonstrates a high precision rate for intrusion detection, achieving a precision rate of 90.53%. In contrast, CNN-IDS, LSTM-IDS, SPARK-IDS, GASVM-IDS, and DDQN-IDS achieve precision rates of 58.89%, 90.27%, 85.85%, 82.64%, and 87.75%, respectively.

The Recall of proposed and existing methods on the CIC-IDS-2017 dataset is shown in Figure 14. Compared with other methods, the proposed HPPELM demonstrates a high recall rate for intrusion detection, achieving a recall rate of 81.59%. In contrast, CNN-IDS, LSTM-IDS, SPARK-IDS, GASVM-IDS, and DDQN-IDS achieve recall rates of 73.68%, 34.63%, 81.24%, 73.74%, and 72.66%, respectively.

Figure 15 shows the FPR of the proposed and existing methods on the CIC-IDS-2017 dataset. Compared with other methods, the proposed HPPELM demonstrates a notably lower false positive rate for intrusion detection, achieving a false positive rate of 2.1%. In contrast, CNN-IDS, LSTM-IDS, SPARK-IDS, GASVM-IDS, and DDQN-IDS achieve false positive rates of 12.62%, 0.91%, 3.28%, 3.8%, and 2.48%, respectively.

Figure 16 shows the F1-score of the proposed and existing methods on the CIC-IDS-2017 dataset. Compared with other methods, the proposed HPPELM demonstrates a high F1-score for intrusion detection, achieving an F1-score of 85.83%. In contrast, CNN-IDS, LSTM-IDS, SPARK-IDS, GASVM-IDS, and DDQN-IDS achieve the F1-score of 65.46%, 50.05%, 83.48%, 77.94%, and 79.49%, respectively.

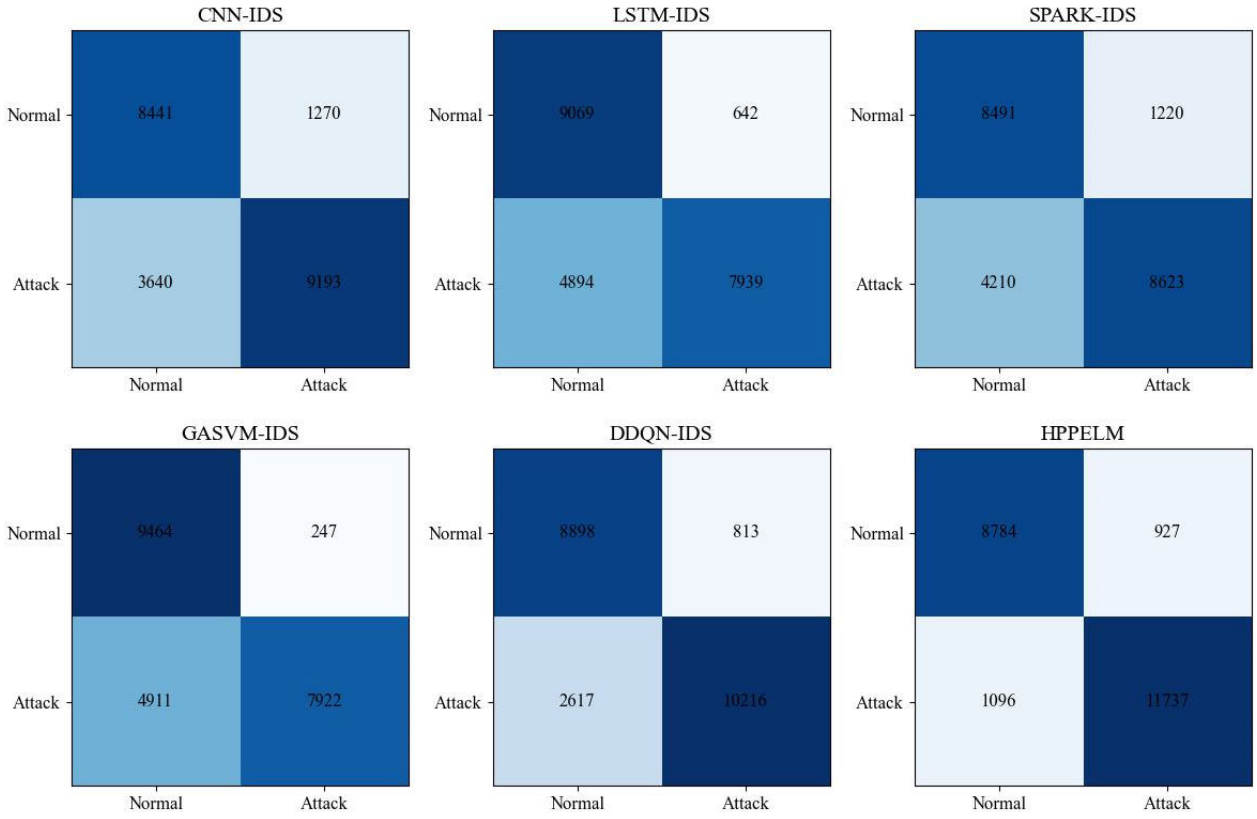


FIGURE 3. Confusion matrix on the NSL-KDD dataset.

TABLE 8. The changes of the fitness values of the DPSO algorithm on the CIC-IDS-2017 dataset.

Model	Accuracy (%)	Precision (%)	Recall (%)	FPR (%)	F1-score (%)	Detection time (s)
HPPELM	94.67	89.59	82.25	2.23	85.83	1.27

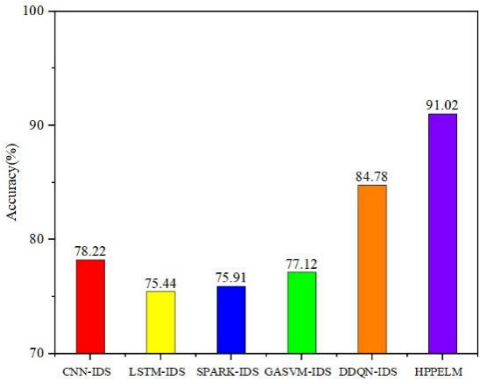


FIGURE 4. Comparison of accuracy on the NSL-KDD dataset.

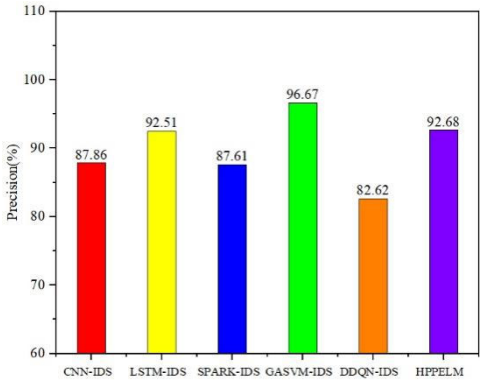


FIGURE 5. Comparison of precision on the NSL-KDD dataset.

Figure 17 shows the Detection time of the proposed and existing methods on the CIC-IDS-2017 dataset. The proposed HPPELM requires a shorter intrusion detection time than other methods. The detection time is 1.27 seconds. In contrast, the detection time for CNN-IDS, LSTM-IDS, SPARK-IDS, GASVM-IDS, and DDQN-IDS are 13.36 seconds, 141.37 seconds, 23.6 seconds, 537.3 seconds, and 412.1 seconds, respectively.

In summary, we achieve good performance in terms of the Accuracy, Precision, Recall, FPR, F1-score, and Detection time through effective feature dimensionality reduction,

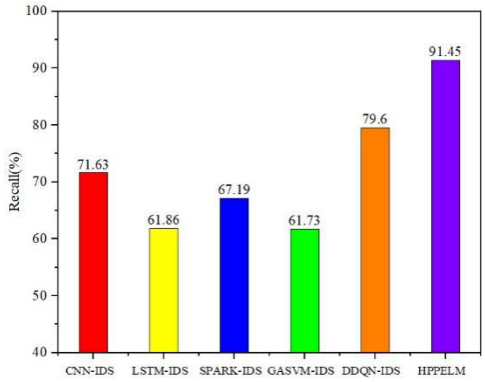


FIGURE 6. Comparison of recall on the NSL-KDD dataset.

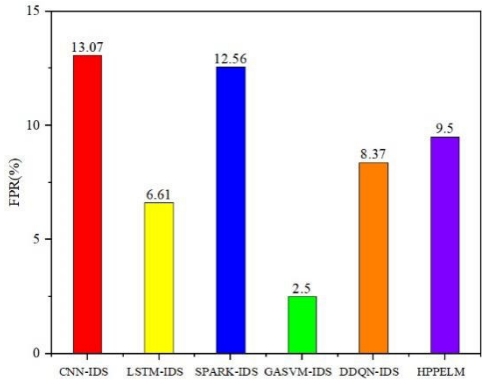


FIGURE 7. Comparison of FPR on the NSL-KDD dataset.

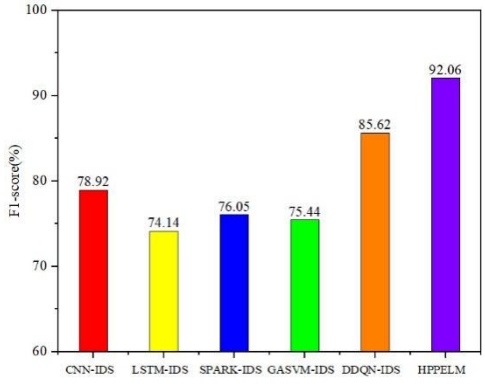


FIGURE 8. Comparison of F1-score on the NSL-KDD dataset.

powerful classification capability, flexible model tuning, and efficient training process.

E. ABLATION ANALYSIS

We further perform ablation experiments on the NSL-KDD and CIC-IDS-2017 datasets to validate the effectiveness of each component within our method. The detailed settings for the ablation experiments are as follows.

1) Compare the WELM with the ELM and analyze its impact on intrusion detection performance.

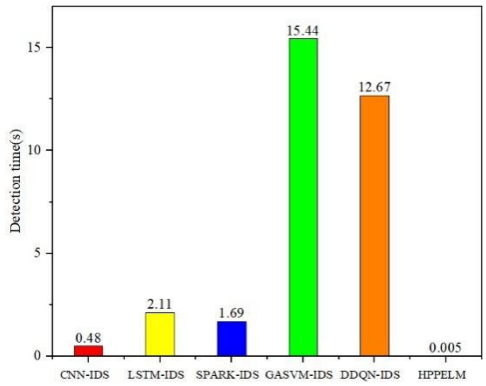


FIGURE 9. Comparison of Detection time on the NSL-KDD dataset.

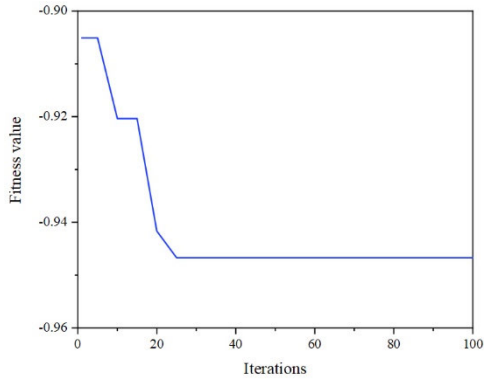


FIGURE 10. The changes of the fitness values of the DPSO algorithm on the CIC-IDS-2017 dataset.

Due to the imbalanced nature of the vehicle network data, there may be significant disparities in the sample counts among various categories. To address this issue, we assign higher weights to essential samples from the minority class. It can help improve the recognition performance for minority categories. The comparison results between WELM and ELM are shown in Table 9.

The data indicate that compared with the ELM, the WELM algorithm is better in overall evaluation indicators.

2) Compare the DPSO with the PSO and analyze its impact on intrusion detection performance.

We investigate the effects of the DPSO and the PSO to assess their impact on finding optimal parameters. The PSO usually uses fixed inertia weights to control the movement of particles, which may lead the algorithm to converge to a locally optimal solution prematurely. We propose the dynamic inertia weight method to adjust the weights flexibly. Table10 presents a comparative analysis of the experimental results between DPSO and PSO.

The results demonstrate that the DPSO enhances the performance of WELM more effectively than the PSO.

3) Compare the GRIPCA with the PCA and analyze its impact on intrusion detection performance.

We introduce the principal component analysis to mitigate the impact of irrelevant or less essential features on

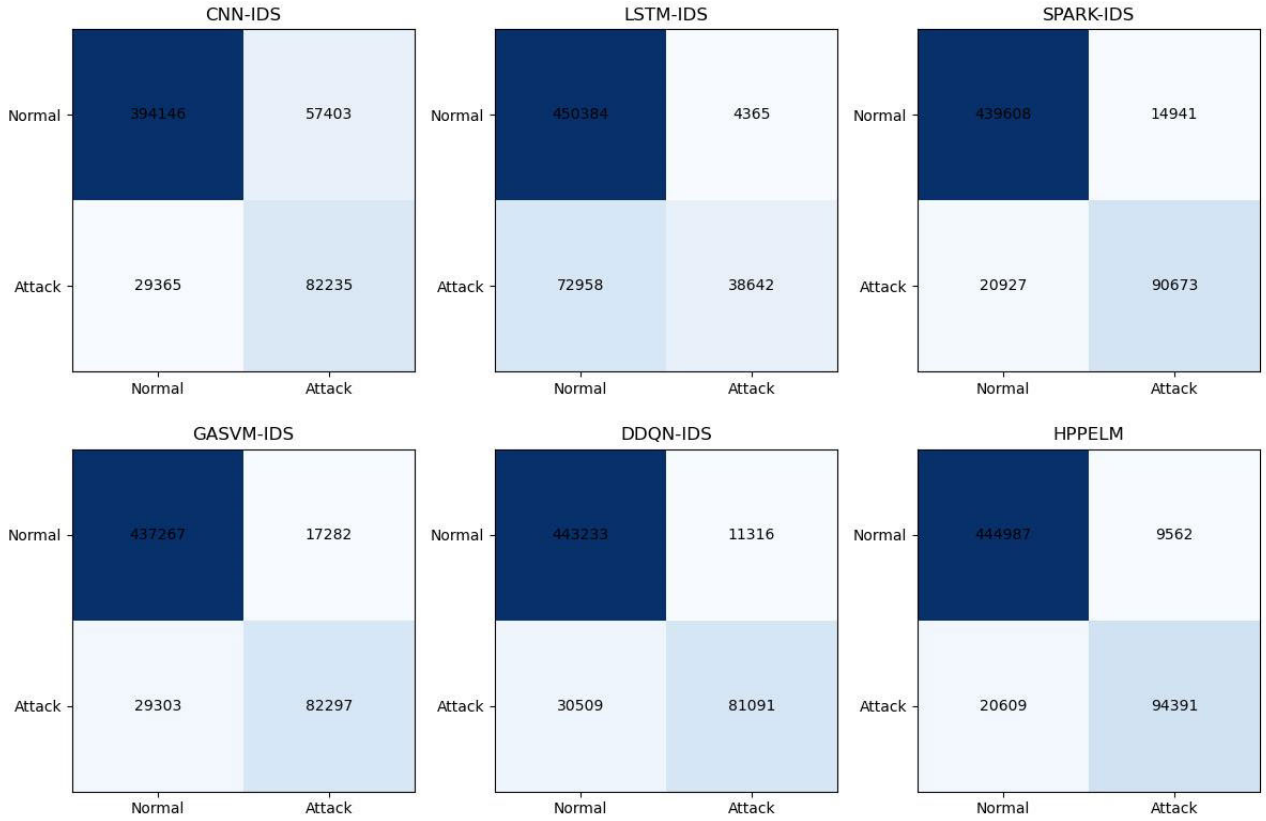


FIGURE 11. Confusion Matrix on the CIC-IDS-2017 dataset.

TABLE 9. Comparison of results between WELM and ELM.

Dataset	Model	Accuracy (%)	Precision (%)	Recall (%)	FPR (%)	F1-score (%)
NSL-KDD	ELM	70.36	78.28	73.73	29.63	69.76
	WELM	74.38	78.23	76.74	25.61	74.28
CIC-IDS-2017	ELM	88.70	92.43	71.95	11.30	77.09
	WELM	91.98	92.47	81.45	8.01	85.55

TABLE 10. Comparison of results between DPSO and PSO.

Dataset	Model	Accuracy (%)	Precision (%)	Recall (%)	FPR (%)	F1-score (%)
NSL-KDD	PSO-WELM	88.91	92.73	87.36	9.04	89.97
	DPSO-WELM	89.77	91.96	89.88	10.37	90.91
CIC-IDS-2017	PSO-WELM	93.19	86.01	78.03	3.09	81.83
	DPSO-WELM	93.69	87.11	79.24	2.77	82.99

TABLE 11. Comparison of results between GRIPCA and PCA.

Dataset	Model	Accuracy (%)	Precision (%)	Recall (%)	FPR (%)	F1-score (%)
NSL-KDD	PCA-DPSO-WELM	84.95	95.72	77.01	4.5	85.35
	HPPELM	91.02	92.68	91.45	9.5	92.06
CIC-IDS-2017	PCA-DPSO-WELM	94.30	89.60	80.05	2.32	84.59
	HPPELM	94.67	89.59	82.25	2.23	85.83

classification outcomes and improve computational efficiency. We conducted a comparative experiment between GRIPCA and PCA, and the results are shown in Table 11.

TABLE 12. The results of combining the components.

Model	Algorithm			NSL-KDD	CIC-IDS-2017
	GRIPCA	DPSO	WELM	Accuracy (%)	
WELM			✓	74.28	91.98
GRIPCAWELM	✓		✓	74.79	92.84
DPSOWELM		✓	✓	89.77	93.69
HPPELM	✓	✓	✓	91.02	94.67

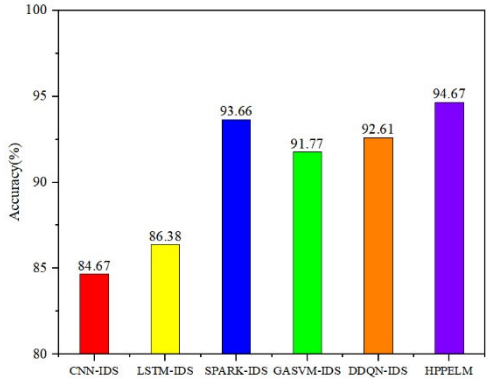


FIGURE 12. Comparison of accuracy on the CIC-IDS-2017 dataset.

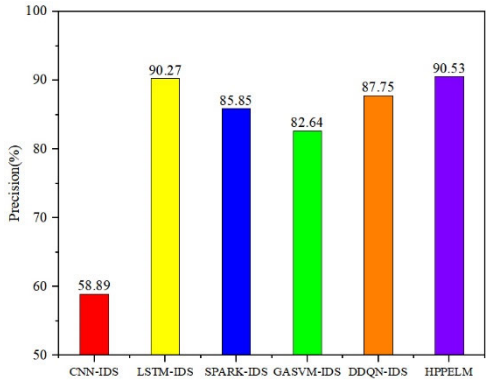


FIGURE 13. Comparison of Precision on the CIC-IDS-2017 dataset.

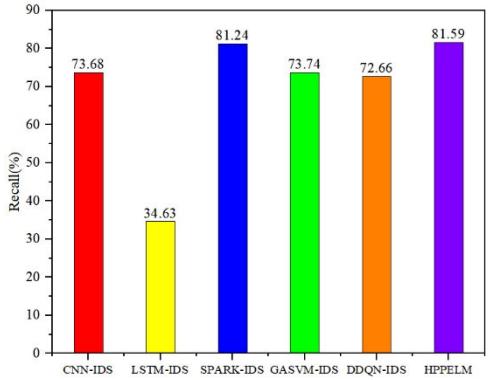


FIGURE 14. Comparison of recall on the CIC-IDS-2017 dataset.

The results demonstrate that the performance of HPPELM is better.

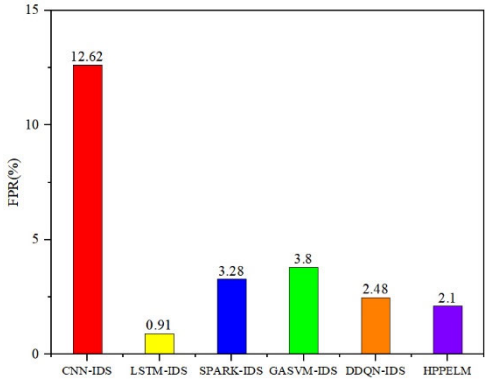


FIGURE 15. Comparison of FPR on the CIC-IDS-2017 dataset.

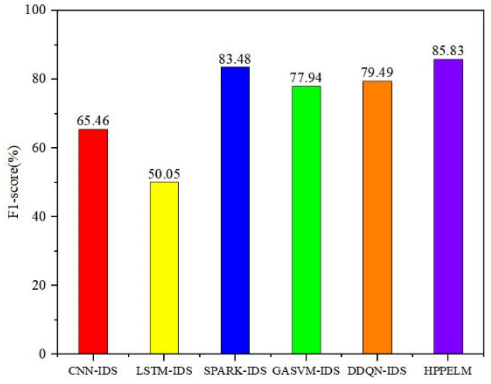


FIGURE 16. Comparison of F1-score on the CIC-IDS-2017 dataset.

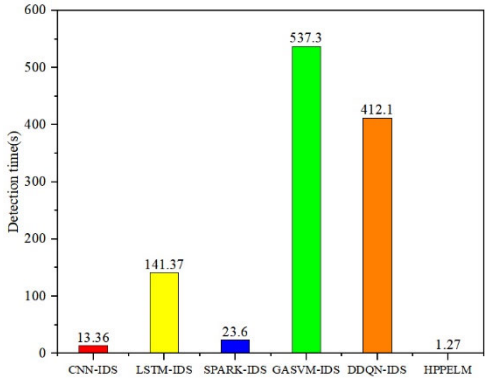


FIGURE 17. Comparison of Detection time on the CIC-IDS-2017 dataset.

4) Combine each component separately for experiments and analyze its impact on intrusion detection performance.

We carry out experiments by combining various components separately, and the results are presented in Table 12.

V. CONCLUSION

In order to enhance the security of Internet of Vehicles networks, we propose an intrusion detection model for Internet of Vehicles using GRIPCA and OWELM. In response to the challenge of limited computing resources in the Internet of Vehicles, we present a solution using GRIPCA. Additionally, we introduce an OWELM for efficiently detecting intrusion attacks in the Internet of Vehicles network. We utilize the NSL-KDD and CIC-IDS-2017 datasets to conduct experiments and compare them with other technologies. The experimental results illustrate that the proposed model excels in multiple metrics. In the future, we intend to employ synthetic minority oversampling technology to randomly oversample the unbalanced data within the Internet of Vehicles to balance the samples and improve the capability of the model to identify minority categories.

REFERENCES

[1] C. Chen, C. Wang, T. Qiu, M. Atiquzzaman, and D. O. Wu, "Caching in vehicular named data networking: Architecture, schemes and future directions," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2378–2407, 4th Quart., 2020, doi: 10.1109/COMST.2020.3005361.

[2] Z. Lv, D. Chen, and Q. Wang, "Diversified technologies in Internet of Vehicles under intelligent edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2048–2059, Apr. 2021, doi: 10.1109/TITS.2020.3019756.

[3] H. Gao, D. Fang, J. Xiao, W. Hussain, and J. Y. Kim, "CAMRL: A joint method of channel attention and multidimensional regression loss for 3D object detection in automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8831–8845, Aug. 2023, doi: 10.1109/TITS.2022.3219474.

[4] L. Xing, P. Zhao, J. Gao, H. Wu, and H. Ma, "A survey of the social Internet of Vehicles: Secure data issues, solutions, and federated learning," *IEEE Intell. Transp. Syst. Mag.*, vol. 15, no. 2, pp. 70–84, Mar. 2023, doi: 10.1109/MITS.2022.3190036.

[5] F. Lone, H. K. Verma, and K. P. Sharma, "A systematic study on the challenges, characteristics and security issues in vehicular networks," *Int. J. Pervasive Comput. Commun.*, vol. 20, no. 1, pp. 56–98, Jan. 2024, doi: 10.1108/ijpcc-04-2022-0164.

[6] S. Xu, Y. Qian, and R. Q. Hu, "Data-driven edge intelligence for robust network anomaly detection," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 1481–1492, Jul. 2020, doi: 10.1109/TNSE.2019.2936466.

[7] F. Fakhfakh, M. Tounsi, and M. Mosbah, "Cybersecurity attacks on CAN bus based vehicles: A review and open challenges," *Library Hi Tech*, vol. 40, no. 5, pp. 1179–1203, Nov. 2022, doi: 10.1108/lht-01-2021-0013.

[8] L. Lihua, "Energy-aware intrusion detection model for Internet of Vehicles using machine learning methods," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–8, May 2022, doi: 10.1155/2022/9865549.

[9] I. Aliyu, S. Van Engelenburg, M. B. Mu'Azu, J. Kim, and C. G. Lim, "Statistical detection of adversarial examples in blockchain-based federated forest in-vehicle network intrusion detection systems," *IEEE Access*, vol. 10, pp. 109366–109384, 2022, doi: 10.1109/ACCESS.2022.3212412.

[10] G. O. Anyanwu, C. I. Nwakanma, J. M. Lee, and D.-S. Kim, "Novel hyper-tuned ensemble random forest algorithm for the detection of false basic safety messages in Internet of Vehicles," *ICT Exp.*, vol. 9, no. 1, pp. 122–129, Feb. 2023, doi: 10.1016/j.icte.2022.06.003.

[11] A. Alsarhan, M. Alauthman, E. Alshdaifat, A.-R. Al-Ghuwairi, and A. Al-Dubai, "Machine learning-driven optimization for SVM-based intrusion detection system in vehicular ad hoc networks," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 5, pp. 6113–6122, May 2023, doi: 10.1007/s12652-021-02963-x.

[12] P. Rani and R. Sharma, "Intelligent transportation system for Internet of Vehicles based vehicular networks for smart cities," *Comput. Electr. Eng.*, vol. 105, Jan. 2023, Art. no. 108543, doi: 10.1016/j.compeleceng.2022.108543.

[13] Z. Li, Y. Kong, and C. Jiang, "A transfer double deep Q network based DDoS detection method for Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 72, no. 4, pp. 5317–5331, Apr. 2023, doi: 10.1109/TVT.2022.3233880.

[14] Y. Wang, G. Qin, M. Zou, Y. Liang, G. Wang, K. Wang, Y. Feng, and Z. Zhang, "A lightweight intrusion detection system for Internet of Vehicles based on transfer learning and MobileNetV2 with hyperparameter optimization," *Multimedia Tools Appl.*, pp. 1–23, Jun. 2023, doi: 10.1007/s11042-023-15771-6.

[15] L. Nie, Z. Ning, X. Wang, X. Hu, J. Cheng, and Y. Li, "Data-driven intrusion detection for intelligent Internet of Vehicles: A deep convolutional neural network-based method," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2219–2230, Oct. 2020, doi: 10.1109/TNSE.2020.2990984.

[16] J. Yang, J. Hu, and T. Yu, "Federated AI-enabled in-vehicle network intrusion detection for Internet of Vehicles," *Electronics*, vol. 11, no. 22, p. 3658, Nov. 2022, doi: 10.3390/electronics11223658.

[17] I. Ahmed, G. Jeon, and A. Ahmad, "Deep learning-based intrusion detection system for Internet of Vehicles," *IEEE Consum. Electron. Mag.*, vol. 12, no. 1, pp. 117–123, Jan. 2023, doi: 10.1109/MCE.2021.3139170.

[18] H. Grover, T. Alladi, V. Chamola, D. Singh, and K. R. Choo, "Edge computing and deep learning enabled secure multitier network for Internet of Vehicles," *IEEE Internet Things J.*, vol. 8, no. 19, pp. 14787–14796, Oct. 2021, doi: 10.1109/JIOT.2021.3071362.

[19] R. Hu, Z. Wu, Y. Xu, and T. Lai, "Vehicular-Network-Intrusion detection based on a mosaic-coded convolutional neural network," *Mathematics*, vol. 10, no. 12, p. 2030, Jun. 2022, doi: 10.3390/math10122030.

[20] L. Yang and A. Shami, "A transfer learning and optimized CNN based intrusion detection system for Internet of Vehicles," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Seoul, South Korea, May 2022, pp. 2774–2779, doi: 10.1109/ICC45855.2022.9838780.

[21] A. Alferaidi, K. Yadav, Y. Alharbi, N. Razmjooy, W. Viriyasitavat, K. Gulati, S. Kautish, and G. Dhiman, "Distributed deep CNN-LSTM model for intrusion detection method in IoT-based vehicles," *Math. Problems Eng.*, vol. 2022, pp. 1–8, Mar. 2022, doi: 10.1155/2022/3424819.

[22] L. Xing, K. Wang, H. Wu, H. Ma, and X. Zhang, "Intrusion detection method for Internet of Vehicles based on parallel analysis of spatio-temporal features," *Sensors*, vol. 23, no. 9, p. 4399, Apr. 2023, doi: 10.3390/s23094399.

[23] S. Panda, S. Rass, S. Moschoyiannis, K. Liang, G. Loukas, and E. Panaousis, "HoneyCar: A framework to configure honeypot vulnerabilities on the Internet of Vehicles," *IEEE Access*, vol. 10, pp. 104671–104685, 2022, doi: 10.1109/ACCESS.2022.3210117.

[24] A. Haydari and Y. Yilmaz, "RSU-based online intrusion detection and mitigation for VANET," *Sensors*, vol. 22, no. 19, p. 7612, Oct. 2022, doi: 10.3390/s22197612.

[25] Y. Zhao, Y. Xun, and J. Liu, "ClockIDS: A real-time vehicle intrusion detection system based on clock skew," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 15593–15606, Sep. 2022, doi: 10.1109/JIOT.2022.3151377.

[26] J. Zhang, B. Gong, M. Waqas, S. Tu, and S. Chen, "Many-objective optimization based intrusion detection for in-vehicle network security," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 15051–15065, Dec. 2023, doi: 10.1109/TITS.2023.3296002.

[27] T. Yu, J. Hu, and J. Yang, "Intrusion detection in intelligent connected vehicles based on weighted self-information," *Electronics*, vol. 12, no. 11, p. 2510, Jun. 2023, doi: 10.3390/electronics12112510.

[28] J. Ahmad, S. A. Shah, S. Latif, F. Ahmed, Z. Zou, and N. Pitropakis, "DRaNN_PSO: A deep random neural network with particle swarm optimization for intrusion detection in the Industrial Internet of Things," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8112–8121, Nov. 2022, doi: 10.1016/j.jksuci.2022.07.023.

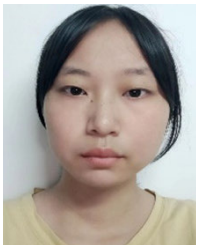
[29] H. Taslimasa, S. Dadkhah, E. C. P. Neto, P. Xiong, S. Ray, and A. A. Ghorbani, "Security issues in Internet of Vehicles (IoV): A comprehensive survey," *Internet Things*, vol. 22, Jul. 2023, Art. no. 100809, doi: 10.1016/j.iot.2023.100809.

[30] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Ottawa, ON, Canada, Jul. 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.

[31] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 108–116, doi: 10.5220/0006639801080116.

[32] T. Wisanwanichthan and M. Thammawichai, "A double-layered hybrid approach for network intrusion detection system using combined naive Bayes and SVM," *IEEE Access*, vol. 9, pp. 138432–138450, 2021, doi: 10.1109/ACCESS.2021.3118573.

KAIJUN ZHANG is currently pursuing the Graduate degree with the School of Cyberspace Security (School of Cryptology), Hainan University. His research interests include network security and AI security.



JIAYU YANG is currently pursuing the bachelor's degree with the School of Computer Science and Technology, Hainan University. Her research interests include blockchain cyber security and AI security.

YANGFEI SHAO is currently pursuing the bachelor's degree with the School of Information and Communication Engineering, Hainan University. Her research interests include blockchain cyber security and AI security.

LEHUA HU received the master' s degree in computer application technology from Kunming University of Science and Technology, in 2013. He is currently an Associate Professor with the School of Information Engineering, Hunan University of Science and Engineering. His research interests include blockchain, image processing, and information security.

WEI OU received the M.S. and Ph.D. degrees from the National University of Defense Technology, in 2005 and 2013, respectively. Since 2020, he has been an Associate Professor with Hainan University. His research interests include cryptography, cyber security, AI security, and blockchain technology.

WENBAO HAN received the M.S. and Ph.D. degrees from Sichuan University, in 1988 and 1994, respectively. Since 2020, he has been a Professor with Hainan University. His research interests include cryptography, cyber security, and AI security.

QIONGLU ZHANG received the Ph.D. degree in communication and information system from the University of Chinese Academy of Sciences, China, in 2021. She is currently a Senior Engineer with the Institute of Information Engineering, Chinese Academy of Sciences, China. Her research interests include data security, cryptography application, the IoT security, and blockchain.