Article citation info:
Dong Z, Wu C. Convolutional neural networks in fault diagnosis of industrial internet of things. Diagnostyka. 2024;25(4):2024406. https://doi.org/ 10.29354/diag/192179.

DIAGNOSTYKA, 2024, Vol. 25, No. 4

e-ISSN 2449-5220
DOI: 10.29354/diag/192179

CONVOLUTIONAL NEURAL NETWORKS IN FAULT DIAGNOSIS OF INDUSTRIAL INTERNET OF THINGS

Zhenzhen DONG *, **Changjie WU**

School of Electronic Information Engineering, Henan Polytechnic Institute, Nanyang, 473000, China
* Corresponding author, e-mail: dongzhenzhen020@163.com

Abstract

Benefiting from the rapid development of Internet technology and communication technology, the Internet of Things industry has risen rapidly. With the rapid development of Internet technology, network security has become increasingly prominent. Moreover, intrusion attacks can cause system failures or reduce system performance, so intrusion detection is an important aspect of ensuring system reliability. Aiming at the great security risks faced by industrial Internet of Things during operation, this study proposes an industrial Internet of Things fault detection model based on a convolutional neural network, which initially screens the intrusion attacks by convolutional neural network, and introduces a particle swarm optimization algorithm to identify the screened intrusion attacks. The experimental results demonstrated that when the training set size was 1600, the accuracy rates of random forest, K-mean clustering algorithm, convolutional neural network and improved convolutional neural network algorithms were 93.2%, 94.9%, 96.3%, and 98.6%, respectively, and the false alarm rates were 6.9%, 5.0%, 3.8%, and 2.1%, respectively. The random forest, K-mean clustering, convolutional neural network, and improved convolutional neural network algorithms had root mean square error values of 0.32, 0.22, 0.18, and 0.11, respectively. The corresponding F1 values were 0.81, 0.84, 0.87, and 0.98 when the training set size was 800. The results of the study demonstrate that the improved algorithmic model outperforms the other strategies, offering a solid foundation for application in the industrial Internet of things.

Keywords: support vector machines, convolutional neural networks, intrusion detection, particle swarm optimization, industrial internet of things

1. INTRODUCTION

The networking and intelligence of production equipment have emerged as a significant trend in the industrial field, driven by the rapid development of industrial Internet of Things (IIoT) technologies. Due to the development of the Internet, the external situation that exists in the interconnected network is also subjected to a huge number of network attacks, moreover, the methods of attacks are endless, and the types of intrusions that need to be resisted are also more and more complicated and diverse [1]. Traditional intrusion detection system (IDS) has several shortcomings, including the need to manually define a large number of rules and features, which limits their effectiveness and reliability in dealing with increasingly complex and changing network environments, and makes the system ineffective in dealing with new and unknown intrusion techniques [2]. Meanwhile, the popularization of the network leads to the expansion of the scale and the complexity of the network environment further becomes higher, and these problems make the accuracy of IDS and the ability to cope with the complexity of the environment particularly important. To address this issue, this research proposes a convolutional neural networks (CNN)-based IIoT fault detection model, which initially screens the intrusion attacks through CNNs and introduces the particle swarm optimization (PSO) algorithm to identify the screened intrusion attacks. It aims to provide theoretical and practical support to improve network security protection. The research contains four main parts. The first part is a brief description of other scholars' research topics on IIoT fault diagnosis. The second part is a review of the main methods used in this research. The model results that are obtained by using the approaches in the research and assessing the findings are presented in the third part. The fourth part is a summary of all the above studies and an outlook for future research.

IIoT being a part of internet is also subjected to frequent intrusion attacks. A cognitive computing-based IDS was proposed by Althobaiti et al. to secure industrial network physical systems. The method used a gated recursive unit model and binary

Received 2024-04-17; Accepted 2024-08-09; Available online 2024-10-25

bacterial foraging optimization to pick features and identify intrusions. The study's findings demonstrated that the suggested model improved the detection rate by 98.45% on industrial network-physical system data [3]. To increase the detection accuracy of IDS, Karthic and Kumar suggested a feature selection technique based on enhanced conditional random field mixed with optimized hybrid deep neural network for classification. The outcomes showed that, in comparison to existing machine learning algorithms, the suggested algorithm performed better on the NSL-KDD and UNSW-NB15 datasets and had higher accuracy [4]. To solve the security challenges in cloud computing, Devi and Muthusenthil suggested a collaborative, distributed, and data-driven intrusion detection and prevention system. The system was built with the intention of providing comprehensive ID for all cloud service providers by leveraging cloud resources. The study's findings demonstrated that the suggested framework could successfully raise cloud computing security and lower infiltration risk [5]. To improve network security of industrial control systems, Zhang et al. suggested a defense in depth concept-based network attack detection system. The study's findings demonstrated that the suggested detection system could successfully identify network intrusions and carry out the early warning role [6].

A local anomaly factor-based ID technique was presented by Ning et al. to address the issue of controller LAN buses without security protection mechanisms. Without changing the protocol or adding to the computational load, the technique made use of the physical properties of the voltage signals on the controller LAN bus to increase the detection accuracy and lower the false detection rate. The research results indicated that the method had good performance and feasibility, and was a powerful means to protect the security of automotive networks [7]. Chen et al. addressed the challenges of conventional machine learning-based IDS in preprocessing and fusion training of heterogeneous data. According to the study's findings, the plan could resolve the cold-start issue, enhance detection accuracy, and streamline the data normalization procedure [8]. A model-agent-based method was presented by Haffar et al. to explain the predictions of black box deep learning models, with the goal of enhancing the interpretability of artificial intelligence. In federated learning scenarios, the approach could be used to identify security and privacy breaches and provide an explanation of their origin. The study's findings showed that the approach can identify attacks with a high degree of accuracy and explain where they originated [9].

In summary, many scholars have already studied ID and achieved certain results, but no scholars have studied ID in IIoT. This research proposes a CNN-based IIoT fault detection model, which initially screens the intrusion attacks by CNN network and introduces PSO algorithm to recognize the screened intrusion attacks.

2. METHODS

The first part analyzes the problems of IIoT and IDS and proposes an offline ID model based on CNN. The second part improves the offline ID model and proposes an online ID model for networks in real environments.

2.1. Offline intrusion detection model based on improved CNNs

IIoT is a type of ubiquitous network that applies Internet of Things (IoT) technology in the industrial sector. IIoT is the application of Internet and sensor technologies to industrial production and manufacturing processes in order to achieve intelligent, real-time and seamless connectivity between devices, systems and people [10]. IIoT aims to boost monitoring and control of industrial processes, lower costs, improve product quality, and increase productivity by integrating a range of sensors, devices, and data analytics. As illustrated in Fig. 1, the IIoT's physical-logical architecture can be split into three sub-layers from bottom to top.

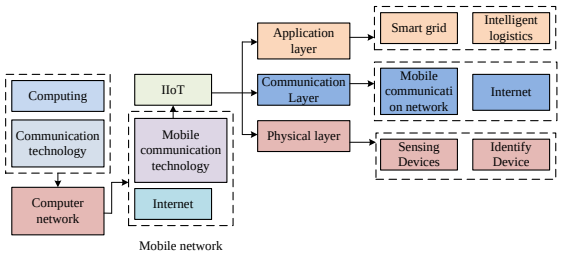


Fig. 1. Structure of IIoT

Fig. 1 depicts the application, communication, and physical layers that make up the IIoT architecture. The physical layer is the lowest layer in the IIoT architecture, which mainly involves hardware and transmission media, including sensors, actuators, IoT devices, communication cables, and network devices. The physical layer is responsible for realizing the transmission and connection of data and ensuring that devices can communicate with each other. The communication layer sits on top of the physical layer and is responsible for handling the transmission of data, communication protocols, and network structure. The communication layer ensures that IoT devices are able to exchange information effectively with each other. The application layer is the highest layer in the IIoT architecture and this layer focuses on processing and implementing specific business applications. On this application layer, data is transmitted from the physical and communication layers, interpreted and processed to provide meaningful information and functionality to the business and users [11]. Among the three structures, the communication layer is the most standardized, industrialized and mature part of the three layers of the IIoT, making the whole IIoT constitute a cooperative and perceptive network. Due to the development of the Internet, the external

situation that exists in the interconnected network also suffers from massive cyber attacks. The attack methods are endless, and the types of invasions that need to be resisted are also more and more complicated and diversified. To address this problem, IDS is proposed, which is a combination of hardware and software that can carry out ID to protect the user's network security. Its workflow is shown in Fig. 2.

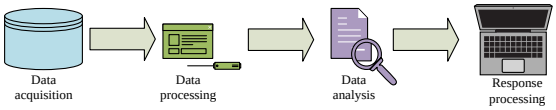


Fig. 2. Workflow diagram of intrusion detection systems

In Fig. 2, relevant information is first collected on the network or in the system log. Then the collected information is preprocessed to integrate the data. To ascertain whether the system has been compromised, the data is then examined and contrasted with the intrusion behaviors found in the database. Finally if the intrusion operation is detected, it responds to the intrusion operation, such as alarm isolation and other anti-intrusion operations. The impact of IDS is contingent upon the quality and quantity of the data collected. It can be reasonably assumed that the more data that is collected, the more effective the subsequent ID will be. The data analysis phase is the core of the whole IDS, through which the trained model is ID, and the model's goodness directly determines the performance of IDS [12]. The traditional IDS is not able to cope with the more complex network environment nowadays. This research proposes an ID method based on CNN and support vector machine (SVM). Fig. 3 illustrates this method's structure.

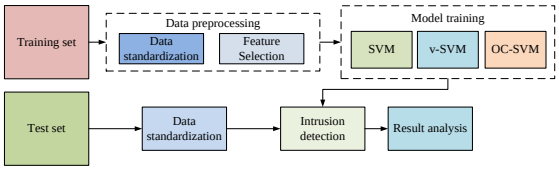


Fig. 3. Structure of intrusion detection model

In Fig. 3, the training set data is then preprocessed to create a hybrid model that combines standard SVMs, v-SVMs, and one-class SVMs. The trained model is then utilized for identification. The majority of the data in the original dataset require preprocessing. Equation (1) illustrates the processing expression.

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

In Equation (1), x denotes the original data and x' is the processed data. max and min are the maximum and minimum value, respectively. After completing the data preprocessing, feature selection is required, which selects a subset from the preprocessed data without changing the representation of the data to be used in the modeling process. The most representative or pertinent features are typically found in the chosen subset, which enhances model performance, lowers dimensionality, lowers the chance of overfitting, and expedites training [13]. Pearson's correlation coefficient, whose expression is given in Equation (2), is used to downscale the data.

$$cor = \frac{COV(X,Y)}{\sigma_Y \sigma_X} = \frac{E((Y-\mu_Y)(X-\mu_X)}{\sigma_Y \sigma_X} \tag{2}$$

In Equation (2), X and Y denote two random features in the data set. μ is the mean of the features in all the data and σ is the standard deviation of the features in all the data. The reduced-dimensional data should be inputted into the model for training purposes. Subsequently, the data should be filtered through a CNN. CNN can automatically learn and extract features from data, from low-level to high-level, reducing the complexity of manually designed features. By sharing parameters and using convolutional kernels, CNN can significantly reduce the number of model parameters and improve computational efficiency. Local connections enable CNNs to capture local features and gradually integrate them into global features, thereby ensuring robustness to local changes. Translation invariance is achieved through pooling layers, making the model insensitive to changes in the position of input data. The parallel computing capability and end-to-end learning method of CNN have improved the efficiency of training and inference. Its structure is shown in Fig. 4.

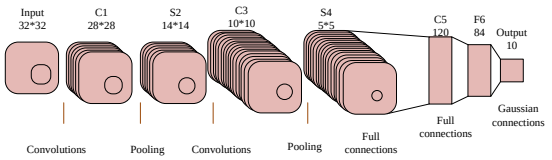


Fig. 4. Structure of CNN

The fundamental function of CNN is convolution. The convolutional layer (CL) uses a convolutional kernel to gradually extract local features from the input data. Equation (3) illustrates the one-dimensional convolution's output formula.

$$C_{cn} = f(X * W_{cn} + b_{cn}) \tag{3}$$

In Equation (3), C_{cn} is the output value. f is the CL activation function and W_{cn} is the weights of the convolutional kernel. X displays the input data and b_{cn} is the bias of the convolution kernel [14]. To achieve the goal of dimensionality reduction, the pooling layer primarily splits the generated feature set. Typically, this is done by using a softmax classifier, which can transfer the output to a normalized probability distribution, or output

confidence, whose expression is given by equation (4).

$$p(x)_i = \frac{e^{z_i}}{\sum_k e^{z_i}}, i = 1, 2, ..., k \tag{4}$$

In Equation (4), k is the classifications, $p(x)_i$ is the confidence level, and z_i is the output value. The data is processed and output into the SVM model and the standard SVM expression is shown in Equation (5).

$$\begin{cases} \sum_{i=1}^{m} a_i y_i = 0 (0 \le a_i \le C, i = 1, 2, \ldots, m) \\ \sum_{i=1}^{m} a_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} a_i a_j y_i \kappa(x_i, x_j) \end{cases} \tag{5}$$

In Equation (5), m is the samples and a denotes the vector corresponding to the samples. When the vector value is greater than 0, the vector is a support vector. C denotes a constant, which takes a value greater than 0. $\kappa(x_i, x_j)$ denotes the kernel function. After CNN completes filtering the data, it is necessary to classify the data to distinguish whether it is an intrusion attack. V-SVM is an improvement of standard SVM, and its basic idea is to place a hyperplane that can separate the data and the origin with the maximum interval value. Firstly, v-SVM introduces the parameter v, which allows users to control the proportion of support vectors and misclassification rates. This provides more flexible model tuning capabilities, allowing users to better balance model complexity and misclassification rate. Secondly, v-SVM has strong robustness and can handle noisy data and outliers, reducing the risk of overfitting and improving the model's generalization ability. In addition, the introduction of the parameter v enables v-SVM to exhibit higher stability when dealing with datasets of different sizes and imbalances. V-SVM can automatically adjust the complexity of the model without manually selecting the value of penalty parameter C, simplifying the parameter tuning process of the model. Its expression is shown in Equation (6).

$$\begin{cases} \min_{\omega, \xi_i, \rho} (\frac{1}{2} \|\omega\|^2 + \frac{1}{vn} \sum_{i=1}^{n} \xi_i - \rho) \\ \left(\omega^T \phi(x_i) \right) > \rho - \xi_i, i = 1, \ldots, n, \xi_i > 0 \end{cases} \tag{6}$$

In Equation (6), n is the samples and ξ_i is the nonzero slack variable penalized in the objective function. ω denotes the hyperplane. ρ denotes the maximum interval value. Then a class of SVMs is trained and finally the intrusion attack is detected.

2.2. Online intrusion detection model based on PSO and CNN

Due to the fact that most of the data in IoT is shared and transmitted in real time, offline intrusion detection based on CNN-SVM has poor performance for IoT. Therefore, an online intrusion detection model based on CNN and PSO algorithm is proposed to replace the SVM structure based on the CNN-SVM model. PSO is an optimization technique that takes its cues from the collective behavior of living things, such as flocks of birds and fish schools [15]. The basic idea behind the method is to mimic the iterative movement of individual particles in the search space in order to identify the best possible answer. The PSO algorithm model's structure is shown in Fig. 4.

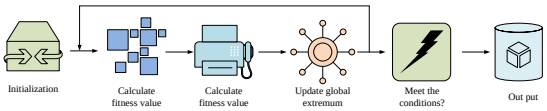


Fig. 5. Flowchart of PSO algorithm

Using random position and velocity, a group of particles representing potential solutions to the problem are first generated in Fig. 5. Each particle is then evaluated in accordance with the problem's objective function to determine its fitness, which is represented by Equation (7).

$$F(X) = \alpha(1 - P) + (1 - \alpha)(1 - \frac{N_f}{N_t}) \tag{7}$$

In Equation (7), α denotes the hyperparameters and P denotes the classifier performance. N_f denotes the subset and N_t denotes the total features. Then for each particle, its individual optimal solution is updated, i.e., the best position reached in its history is recorded. The best adapted particle in the population is selected and its position is used as the optimal solution for the whole population [16]. For each particle, the speed and position are updated according to its individual optimal solution and the group optimal solution. Equation (8) displays the formula for updating it.

$$V_{id}(t+1) = \omega v_{id}(t) + c_2 r_2 (-x_{id}(t) + p_{gd}(t)) + c_1 r_1 (-x_{id}(t) + p_{id}(t)) \tag{8}$$

In Equation (8), c_1 represents the cognitive coefficient, which determines the importance of an individual's optimal position. c_2 represents the social coefficient, which determines the importance of the global optimal position. r_1 represents a random number between 0 and 1, which is used to introduce randomness and ensure search diversity. r_2 represents another random number between 0 and 1. N denotes the total particles in the population and d denotes the dimension. t denotes the iterations, and ω denotes the non-negative inertia factor. p denotes the optimal position. Fig. 5 displays the proposed online ID paradigm.

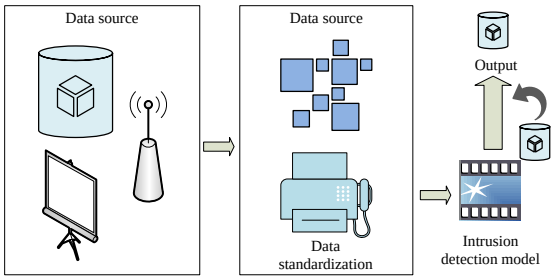


Fig. 6. Online intrusion detection model

The three key components of the online ID model in Fig. 6 are data collecting, data preparation, and model construction. First, the data set is preprocessed; non-numerical data in the original data set must be converted to numerical data because the model requires numerical data for the input [17]. After the data has been feature selected, a gradient boosting decision tree is rapidly constructed using LightGBM and Pearson's correlation coefficient. The optimization goal of the gradient boosting framework is to minimize the loss function (LF). For the regression problem, a typical mean square error LF is shown in Equation (9).

$$L(\theta)=\frac{1}{2N}\sum_{i=1}^{N}(y_i-F(x_i))^2 \tag{9}$$

In Equation (9), N denotes the samples, y_i denotes the true label, and $F(x_i)$ denotes the predicted value (PV). The core of gradient boosting is to update the model parameters by gradient descent method. For the LF, the gradient is usually computed as the partial derivative of the LF with respect to the PV of the model. For the mean square error LF described above, the gradient is calculated as shown in Equation (10).

$$\frac{\partial L}{\partial F(x_i)}=F(x_i)-y_i \tag{10}$$

In Equation (10), $F(x_i)$ is the PV and y_i is the output value. LightGBM uses a histogram gradient boosting algorithm, which hinges on selecting the optimal node splits [18]. The gain of node splitting is shown in Equation (11).

$$Gain=\frac{1}{2}[\frac{Gain_l^2}{H_l+\lambda}+\frac{Gain_r^2}{H_r+\lambda}-\frac{Gain_r^2+Gain_l^2}{H_r+H_l+\lambda}]-\lambda \tag{11}$$

In Equation (11), $Gain$ denotes the gain of the child node, H is the sum of the child node matrix, and λ is the regularization term. After generating the gradient boosting decision tree by LightGBM, the global importance formula of the features is calculated as shown in Equation (12).

$$\begin{cases} J_j^2=\frac{1}{M}\sum_{m=1}^{M}J_J^2(T_m) \\ J_j^2(T)=\sum_{t=1}^{L-1}i_t^2(v_t=j) \end{cases} \tag{12}$$

In Equation (12), J_j^2 denotes the global importance of the computed feature and M denotes the the generated gradient spanning tree. L denotes the leaf nodes and $L-1$ denotes the nodes other than leaf nodes in the generated gradient spanning tree. i_t^2 denotes the amount of reduction of the squared loss obtained after the node completes the node splitting of the generated gradient spanning tree.

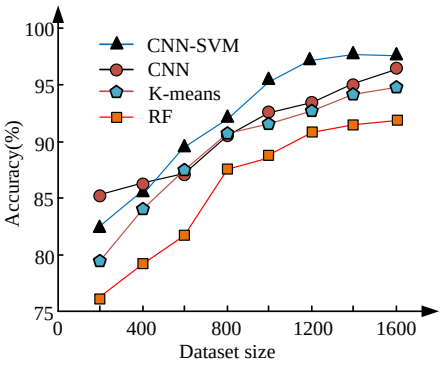
3. RESULTS

In the first part, the K-mean clustering method (K-means) and the random forest algorithm (RF) are introduced. The accuracy and LF measures are employed in these methods to analyze the offline ID model's performance. The second part analyzes the performance of online ID network models.

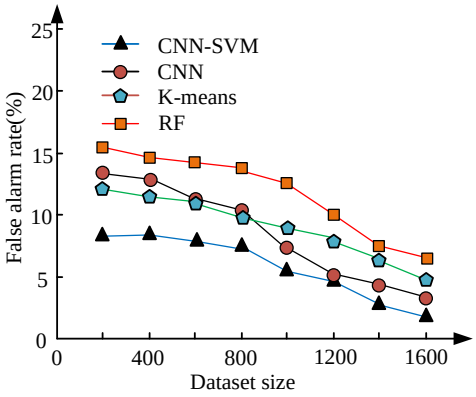
3.1. Offline intrusion detection model based on improved CNNs

The server CPU used in this research is Inter(R) Core(TM) i5-10210U with 16GB of RAM, and the GPU is NVIDIA Geforce GTX2080Ti with 8GB of video memory. The operating system is Windows 10. The dataset used is the UNSW-NB15 dataset, which simulates real-world network activity, including normal traffic and various attacks. The dataset contains normal traffic and nine different types of network attacks, such as fuzzers, analytics, backdoors, denial of service, remote access Trojans, worms, Shellcode, detection, and spyware. In addition, the dataset contains 49 features, including basic features (such as source IP, destination IP, ports, etc.), content features (such as byte count, packet count, etc.), and temporal features (such as traffic duration, etc.), as well as higher-order features based on statistics and information theory. The dataset covers various types of attacks and normal traffic in modern network environments, with high diversity and representativeness. Moreover, it provides rich features that facilitate in-depth research on different detection methods. 2000 data are randomly selected and divided according to the ratio of 4:1 between training set and validation set. To compare this suggested strategy with RF and K-means, the results are displayed in Fig. 7 [19-20].

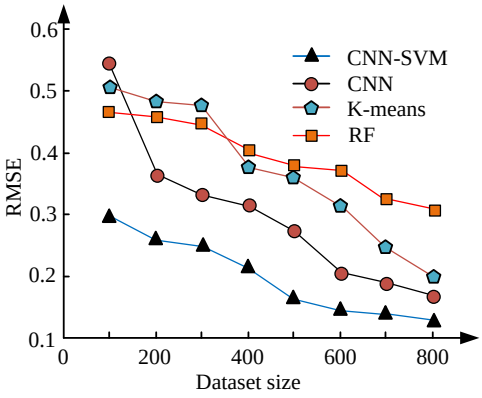
Figs. 7 (a), 7 (b), 7 (c), and 7 (d) respectively represent the accuracy, false alarm rate, RMSE value, and F1 value of each algorithm on different sizes of training sets. From Fig. 7(a), the accuracy of each algorithm model increases as the training dataset increases. At a training set size of 1600, the accuracy of RF, K-means, CNN, and CNN-SVM algorithms are 93.2%, 94.9%, 96.3%, and 98.6%, respectively. In Fig. 7(b), the false alarm rate of each algorithm model is decreasing as the training dataset increases. At a training set size of 1600, the false alarm rates of RF, K-means, CNN, and CNN-SVM algorithms are 6.9%, 5.0%, 3.8%, and 2.1%, respectively. In Fig. 7(c), the RMSE values of each algorithm model are decreasing as the training data set increases. When the training set size is 800, the



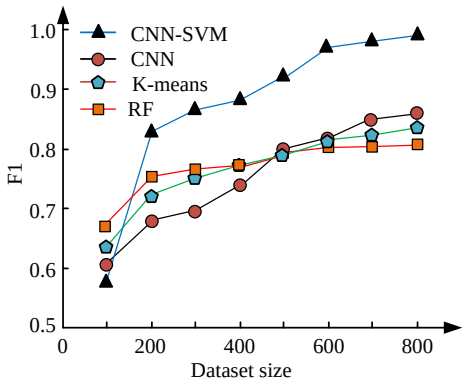
(a) Comparison of accuracy of different algorithms



(b) Comparison of false alarm rate of different algorithms



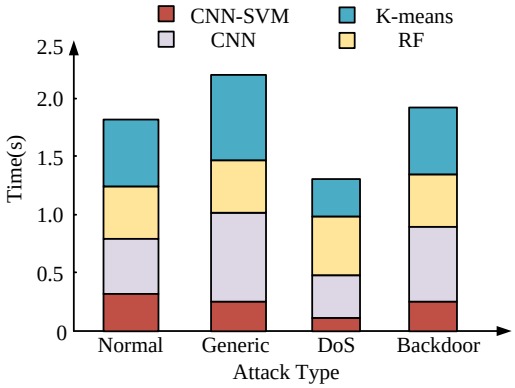
(c) Comparison of RMSE values between different models



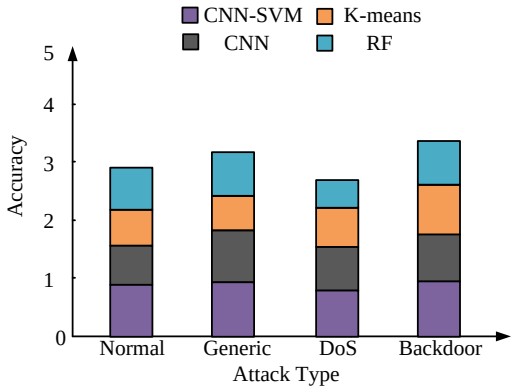
(d) Comparison of F1 between different models

Fig. 7. Performance comparison of four algorithmic models

RMSE values of RF, K-means, CNN, and CNN-SVM algorithms are 0.32, 0.22, 0.18, and 0.11, respectively. In Fig. 7(d), as the training dataset size increases, the F1 value of each algorithmic model is increasing. When the training set size is 800, the F1 values of RF, K-means, CNN, and CNN-SVM algorithms are 0.81, 0.84, 0.87, and 0.98, respectively. According to the experimental results, out of the four technique models, the suggested CNN-SVM algorithm performs the best and can attain higher performance with less training data. Five more common intrusion attacks in the dataset are selected to compare the time consumed by the four algorithms for ID, and the results are shown in Fig. 8.



(a) Time under different Attack Type



(b) Accuracy under different Attack Type

Fig. 8. Comparison of model performance against different intrusion attacks

Fig. 8(a) represents the recognition time of different algorithmic models for four different intrusion attack types and Fig. 8(b) represents the recognition accuracy of different algorithmic models for four different intrusion attack types. In Fig. 8(a), among the four intrusion types, the recognition time of each method is longer for Generic intrusion type, in which the detection time of CNN-SVM

algorithmic model for Normal, Generic, DoS, and Backdoor intrusion types are 187ms, 152ms, 68ms, and 156ms, respectively. In Fig. 8(b), the algorithmic model of each recognize the Dos intrusion type correctly is low. Among them, the detection accuracy of CNN-SVM algorithm model for Normal, Generic, DoS and Backdoor intrusion

Table 1. Detection results of each method for different types of intrusion detection

Type	RF			K-means			CNN			CNN-SVM		
	ACC	RMSE	Time (ms)	ACC	RMSE	Time (ms)	ACC	RMSE	Time (ms)	ACC	RMSE	Time (ms)
Generic	76.8	40.9	286	79.4	34.8	212	85.5	19.5	131	90.7	19.5	57
Exploits	78.1	42.2	278	80.7	36.1	204	86.8	20.8	123	92.1	20.8	49
Fuzzers	73.6	37.7	347	76.2	31.6	273	82.3	16.3	192	87.5	16.3	118
DoS	65.8	29.9	369	69.6	23.8	295	75.7	8.5	214	80.9	8.5	140
Reconnaissance	76.6	40.7	297	80.4	34.6	223	86.5	19.3	142	91.7	19.3	68
Analysis	66.8	30.9	284	70.6	24.8	206	76.7	9.5	178	81.9	9.5	125
Backdoor	71.2	35.3	296	75.8	29.2	218	81.9	13.9	190	87.1	13.9	80
Shellcode	78.3	42.7	271	82.1	36.6	193	88.2	21.3	165	93.4	21.3	55
Worms	69.7	34.1	265	73.5	28.7	244	79.6	13.4	216	84.8	13.4	106

types are 0.91, 0.98, 0.79 and 0.99, respectively. The outcomes illustrate that the proposed CNN-SVM algorithm is able to show good performance in different intrusion types. The total performance of each method is compared, and Table 1 shows the results.

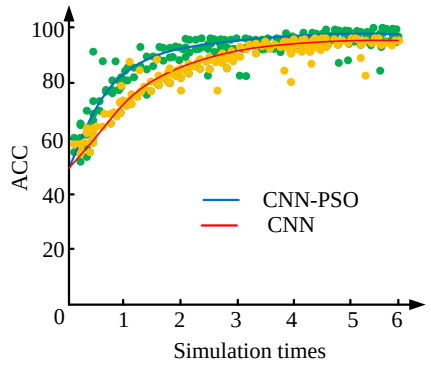
In Table 1, among the nine intrusion types, the individual methods have lower detection accuracy for DoS, Backdoor and Worms intrusion types and higher detection accuracy for Exploits and Shellcode intrusion types. Among the methods, the RF model has the lowest recognition accuracy, the largest RMSE value and the longest detection time. The proposed CNN-SVM algorithm model has the highest recognition accuracy, the smallest RMSE value and the shortest detection time. According to the testing results, out of the four algorithmic models, the suggested CNN-SVM algorithmic model performs the best and is also effective against various kinds of attacks.

3.2. Performance analysis of online intrusion detection model based on PSO and CNN

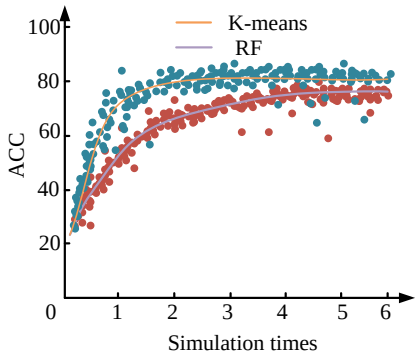
In the networking state of the verification model's performance, nine selected intrusion types are identified as more excellent and more difficult, with two identified for each. The identification performance and simulation of the model are then analyzed. The results are shown in Fig. 9.

The CNN-PSO and CNN algorithm models' classification performance is shown in Fig. 9(a), while the K-means and RF models' classification performance is shown in Fig. 9(b). In this figure, in the CNN-PSO algorithm model, most of its simulation results show more excellent performance. Among the four algorithmic models, the proposed CNN-PSO algorithmic model shows good recognition performance. Fig. 10 presents the results of a comparison of the four approach models' overall performance on the validation set.

The accuracy of each algorithm under various iteration counts is displayed in Fig. 10(a), and the accuracy of each algorithm under various validation sets is displayed in Fig. 10(b). As the iterations increases in Fig. 10(a), the performance of each algorithm model progressively improves. The CNN-PSO algorithm model essentially reaches its peak performance when the number of iterations approaches fifty. When the iterations is 250, the



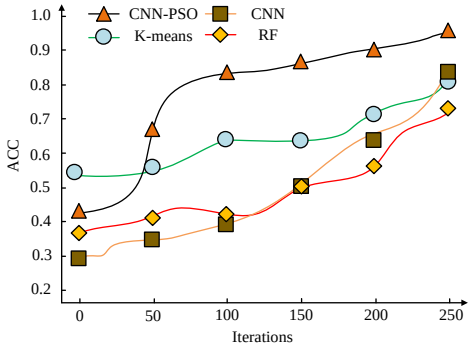
(a) Model information under different simulation times



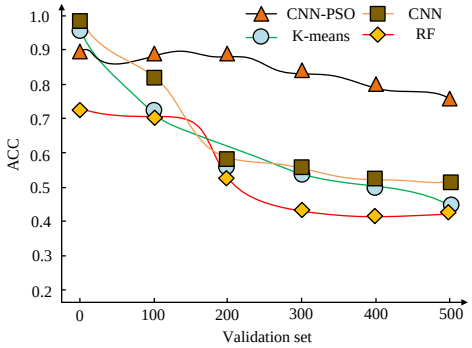
(b) Model information under different simulation times

Fig. 9. Comparison of classification performance of each algorithmic model

accuracy of RF, K-means, CNN, and CNN-PSO algorithm models are 0.73, 0.81, 0.85, and 0.94, respectively. In Fig. 10(b), the performance of each model decreases as the validation set increases, and the accuracy of RF, K-means, CNN, and CNN-PSO algorithm models is 0.73 when the validation set is 500. models have accuracies of 0.42, 0.45, 0.52, and 0.77, respectively. The suggested algorithmic model performs the best out of all the models, according to the experimental data. To rate the models used in the study, fifty participants are chosen at random and placed into five groups. The results are displayed in Table 2.



(a) ACC of various models under different Iterations



(b) ACC of various models under different Validation set sizes

Fig. 10. Performance comparison of different algorithms

Table 2. User evaluation form

Model	Group 1	Group 2	Group 3	Group 4	Group 5
CNN-PSO	91.6	92.7	88.4	89.8	89.9
CNN	85.2	85.7	85.6	83.6	82.5
K-means	82.2	80.6	81.5	84.8	79.5
RF	76.4	78.3	79.8	77.9	76.3

In Table 2, the ratings of the five groups for the CNN-PSO algorithm model are 91.6, 92.7, 88.4, 89.8, and 89.9, respectively. The ratings for the CNN algorithm model are 85.2, 85.7, 85.6, 83.6, and 82.5, respectively. The ratings for the K-means algorithm model are 82.2, 80.6, 81.5, and 84.8, respectively, 79.5. The ratings for the RF model are 76.4, 78.3, 79.8, 77.9, and 76.3, respectively. The trial findings demonstrate how well-liked the suggested CNN-PSO algorithm model is among consumers.

4. CONCLUSION

With the development of the Internet, IIoT has been applied to various fields. To address the problem that IoT is often subject to intrusion attacks, this research proposes a CNN-based IIoT fault detection model. It initially screens intrusion attacks by CNN network and introduces PSO algorithm to recognize the screened intrusion attacks. The outcomes demonstrated that the performance of each algorithmic model was enhanced with the increase of the training dataset, and the accuracy of the RF, K-means, CNN, and CNN-SVM algorithms were 93.2%, 94.9%, 96.3%, and 98.6% when the training set size was 1600, and the false alarm rate was 6.9%, 5.0%, 3.8%, and 2.1%, respectively. When the size of the training set was 800, the RMSE values of RF, K-means, CNN, and CNN-SVM algorithms were 0.32, 0.22, 0.18, and 0.11, and the F1 values were 0.81, 0.84, 0.87, and 0.98, respectively. The CNN-SVM algorithm model's detection times for Normal, Generic, DoS, and Backdoor intrusion types were 187ms, 152ms, 68ms, and 156ms, respectively, and the detection accuracy were 0.91, 0.98, 0.79, and 0.99, respectively. When the number of iterations reaches about 50, the performance of the CNN-PSO algorithm model basically reaches the maximum. When the number of iterations was 250, the accuracy of RF, K-means, CNN, and CNN-PSO algorithm models were 0.73, 0.81, 0.85, and 0.94, respectively. The outcomes demonstrate that, out of all the models, the suggested algorithmic model performs the best. Nevertheless, this research is not without limitations. The adopted dataset comprises a smaller number of attack types than would be observed in a real network, where a greater diversity of intrusion attacks is likely to occur. By expanding the range of attacks included in the dataset, it would be possible to more rigorously assess the robustness of the model.

Source of funding: *This research received no external funding.*

Author contributions: *research concept and design, Z.D.; Collection and/or assembly of data, Z.D..; Data analysis and interpretation, C.W.; Writing the article, C.W.; Critical revision of the article, Z.D.; Final approval of the article, Z.D., C.W.*

Declaration of competing interest: *The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.*

REFERENCES

1. Zhou Y, Cheng G, Jiang S, Dai M. Building an efficient intrusion detection system based on feature selection and ensemble classifier. Computer Networks 2020; 174: 107247. https://doi.org/10.1016/j.comnet.2020.107247.
2. Nazir A, Khan RA. A novel combinatorial optimization based feature selection method for network intrusion detection. Computers & Security 2021; 102: 102164. https://doi.org/10.1016/j.cose.2020.102164.
3. Althobaiti MM, Pradeep Mohan Kumar K, Gupta D, Kumar S, Mansour RF. An intelligent cognitive computing based intrusion detection for industrial cyber-physical systems. Measurement 2021; 186: 110145. https://doi.org/10.1016/j.measurement.2021.110145.
4. Karthic S, Kumar SM. Hybrid optimized deep neural network with enhanced conditional random field based intrusion detection on wireless sensor network. Neural

Processing Letters 2023; 55(1): 459–79. https://doi.org/10.1007/s11063-022-10892-9.
5. Devi K, Muthusenthil B. Intrusion detection framework for securing privacy attack in cloud computing environment using DCCGAN-RFOA. Transactions on Emerging Telecommunications Technologies 2022; 33(9): e4561. https://doi.org/10.1002/ett.4561.
6. Zhang F, Kodituwakku HADE, Hines JW, Coble J. Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data. IEEE Transactions on Industrial Informatics 2019; 15(7): 4362–9. https://doi.org/10.1109/TII.2019.2891261.
7. Ning J, Wang J, Liu J, Kato N. Attacker identification and intrusion detection for in-vehicle networks. IEEE Communications Letters 2019; 23(11): 1927–30. https://doi.org/10.1109/LCOMM.2019.2937097.
8. Chen D, Zhang F, Zhang X. Heterogeneous IoT intrusion detection based on fusion word embedding deep transfer learning. IEEE Transactions on Industrial Informatics 2023; 19(8): 9183–93. https://doi.org/10.1109/TII.2022.3227640.
9. Haffar R, Sánchez D, Domingo-Ferrer J. Explaining predictions and attacks in federated learning via random forests. Applied Intelligence 2023; 53(1): 169–85. https://doi.org/10.1007/s10489-022-03435-1.
10. Ge M, Syed NF, Fu X, Baig Z, Robles-Kelly A. Towards a deep learning-driven intrusion detection approach for Internet of Things. Computer Networks 2021; 186: 107784. https://doi.org/10.1016/j.comnet.2020.107784.
11. Mighan SN, Kahani M. A novel scalable intrusion detection system based on deep learning. International Journal of Information Security 2021; 20(3): 387–403. https://doi.org/10.1007/s10207-020-00508-5.
12. Kamaldeep, Malik M, Dutta M, Granjal J. IoT-Sentry: a cross-layer-based intrusion detection system in standardized internet of things. IEEE Sensors Journal 2021; 21(24): 28066–76. https://doi.org/10.1109/JSEN.2021.3124886.
13. Gao B, Bu B, Zhang W, Li X. An intrusion detection method based on machine learning and state observer for train-ground communication systems. IEEE Transactions on Intelligent Transportation Systems 2022; 23(7): 6608–20. https://doi.org/10.1109/TITS.2021.3058553.
14. Saba T, Sadad T, Rehman A, Mehmood Z, Javaid Q. Intrusion detection system through advance machine learning for the internet of things networks. IT Professional 2021; 23(2): 58–64. https://doi.org/10.1109/MITP.2020.2992710.
15. Xie G, Yang LT, Yang Y, Luo H, Li R, Alazab M. Threat analysis for automotive CAN networks: A GAN model-based intrusion detection technique. IEEE Transactions on Intelligent Transportation Systems 2021; 22(7): 4467–77. https://doi.org/10.1109/TITS.2021.3055351.
16. Khalafi ZS, Dehghani M, Khalili A, Sami A, Vafamand N, Dragicevic T. Intrusion detection, measurement correction, and attack localization of PMU networks. IEEE Transactions on Industrial Electronics 2022; 69(5): 4697–706. https://doi.org/10.1109/TIE.2021.3080212.
17. Wang J, Tian Z, Zhou M, Wang J, Yang X, Liu X. Leveraging hypothesis testing for CSI based passive human intrusion direction detection. IEEE Transactions on Vehicular Technology 2021; 70(8): 7749–63. https://doi.org/10.1109/TVT.2021.3090800.
18. Bhosle K, Musande V. Evaluation of deep learning CNN Model for recognition of devanagari digit. Artificial Intelligence and Applications 2023; 1(2): 114–8. https://doi.org/10.47852/bonviewAIA3202441.
19. Hussain K, Xia Y, Onaizah AN, Manzoor T, Jalil K. Hybrid of WOA-ABC and proposed CNN for intrusion detection system in wireless sensor networks. Optik 2022; 271: 170145. https://doi.org/10.1016/j.ijleo.2022.170145.
20. Otair M, Ibrahim OT, Abualigah L, Altalhi M, Sumari P. An enhanced Grey Wolf optimizer based particle swarm optimizer for intrusion detection system in wireless sensor networks. Wireless Networks 2022; 28(2): 721–44. https://doi.org/10.1007/s11276-021-02866-x.

Zhenzhen DONG
In 2006, she obtained a Bachelor's degree in Communication Engineering from the PLA Information Engineering University in Zhengzhou, China. In 2009, she obtained a Master's degree in Communication and Information Systems from Xi'an University of Technology in Xi'an, China. Currently, she is teaching at School of Electronic Information Engineering, Henan Polytechnic Institute in China. Her areas of interest include electronic information, Internet of Things technology, and communication technology.
e-mail: dongzhenzhen020@163.com

Changjie WU
received his Bachelor's degree in Measurement and Control Technology and Instrumentation from BIT, Beijing, China in 2013. He received his Master's degree in Physical Electronics from NCRIEO, Beijing, China in 2017. Presently, he serves as a teacher and conducts research work at Henan Polytechnic Institute, Henan, China. His areas of interest are electronic circuits, Internet of Things technology, and electronic materials.
e-mail: wuchangjie010@163.com