Journal of Information Hiding and Privacy Protection
DOI: 10.32604/jihpp.2022.029922
Article

Tech Science Press

Intrusion Detection System Using a Distributed Ensemble Design Based Convolutional Neural Network in Fog Computing

Aiming Wu[1], Shanshan Tu[1,*], Muhammad Wagas[1,2,3], Yongjie Yang[1], Yihe Zhang[1] and Xuetao Bai[1]

[1]Engineering Research Center of Intelligent Perception and Autonomous Control, Faculty of Information Technology, Beijing University of Technology, Beijing, 100124, China
[2]Faculty of Computer Science and Engineering, GIK Institute of Engineering Sciences and Technology, Topi 23460, Pakistan
[3]School of Engineering, Edith Cowan University, Joondalup Perth, WA 6027, Australia
*Corresponding Author: Shanshan Tu. Email: sstu@bjut.edu.cn
Received: 15 March 2022; Accepted: 05 May 2022

Abstract: With the rapid development of the Internet of Things (IoT), all kinds of data are increasing exponentially. Data storage and computing on cloud servers are increasingly restricted by hardware. This has prompted the development of fog computing. Fog computing is to place the calculation and storage of data at the edge of the network, so that the entire Internet of Things system can run more efficiently. The main function of fog computing is to reduce the burden of cloud servers. By placing fog nodes in the IoT network, the data in the IoT devices can be transferred to the fog nodes for storage and calculation. Many of the information collected by IoT devices are malicious traffic, which contains a large number of malicious attacks. Because IoT devices do not have strong computing power and the ability to detect malicious traffic, we need to deploy a system to detect malicious attacks on the fog node. In response to this situation, we propose an intrusion detection system based on distributed ensemble design. The system mainly uses Convolutional Neural Network (CNN) as the first-level learner. In the second level, the random forest will finally classify the prediction results obtained in the first level. This paper uses the UNSW-NB15 dataset to evaluate the performance of the model. Experimental results show that the model has good detection performance for most attacks.

Keywords: Intrusion detection system; fog computing; convolutional neural network; feature selection

1 Introduction

The Internet of Things is considered to be the next era of communication. Through the Internet of Everything, physical objects can break through the limitations of space to create, receive, and exchange data in a seamless manner [1]. The Internet of Things is constantly evolving, aiming to connect a variety of different intelligent physical objects to realize the modernization of various fields [2]. With

the rapid development of the Internet of Things, Internet of Things devices have been widely used in homes, public infrastructure, enterprises and other fields, so Internet of Things devices will gradually penetrate into people's daily lives [3]. However, IoT systems are vulnerable to various security attacks, such as worms, exploits, denial-of-service (DoS) and backdoor. These attacks will damage a large number of IoT devices, services and smart environment devices [4].

Fog computing was developed by Cisco in 2012 as an extension of the cloud to the edge of the network. It extends the cloud to the edge of the network, thereby providing efficient data access, data storage and data computing. Generally speaking, the goal of fog computing is to reduce data and traffic on cloud servers, reduce latency, and improve service quality. In addition, fog nodes can help IoT devices perform complex and heavy storage and calculations. In this way, IoT devices have outstanding advantages in deploying distributed and parallel security [5].

On the other hand, fog computing can decentralize storage and computing in cloud services to fog nodes, which requires a powerful detection system to protect this IoT system, so an intrusion detection system came into being [6]. The purpose of the intrusion detection system is to detect all kinds of abnormal network traffic and network activities that are not recognized by traditional firewalls. This is of great significance for the safe, complete and confidential operation of the Internet of Things system. Intrusion detection systems can be divided into two types, one is signature-based intrusion detection system (SIDS), and the other is anomaly-based intrusion detection system (AIDS) [7]. When the predefined abnormal network data is detected, the signature-based intrusion detection system (SIDS) will directly detect the intrusion. In fact, in the context of the Internet of Things system, it is actually very unrealistic to rely on predefined attack characteristics. Many attacks have not been seen before, and this also consumes a lot of storage resources. Anomaly-based intrusion detection system (AIDS) analyzes normal network behavior to determine whether the network pattern is considered an intrusion [8].

Generally speaking, the current data sets related to intrusion detection mainly include KDD-Cup99 [9], UNSW-NB15 [10], ADFA-LD (2013) [11], NSL-KDD (2007) [12], ITOC CDX (2009) [13] and gureKddcup (2008) [14]. KDDCup99 is one of the widely used publicly available data sets currently used in intrusion detection systems. However, it has now been discovered that this data set has some inherent problems, and its training set and test set have a large number of redundant and repeated records. These records will bias the learning algorithm and thus mislead the results [12]. So in this paper we propose to use UNSW-NB15 as the dataset for this article. Because it contains modern attacks that KDDCup99 or NSL-KDD does not have, it is more reasonable and more substantial.

In order to defend against various modern Internet of Things attacks, this paper proposes a distributed intrusion detection system based on fog computing environment based on convolutional neural network and random forest integration algorithm. We first train the model through a convolutional neural network. On this basis, the output is used to train the random forest. At this link, the normal records and attack records can be finally classified. In the test, the testing dataset generated by the IoT device cluster is used to finally evaluate the model after training. The distributed detection system is deployed on the fog node and is responsible for monitoring the traffic entering the fog node and judging whether the fog node has been attacked. IoT devices are mainly composed of sensors. The sensors have the same detection radius. A certain number of sensor nodes form a cluster to detect various parameters in the current environment, and then send the collected data to the fog node for processing. The logs of various fog nodes are saved on the cloud server to manage the fog nodes. Fig. 1 shows the entire distributed intrusion detection model.

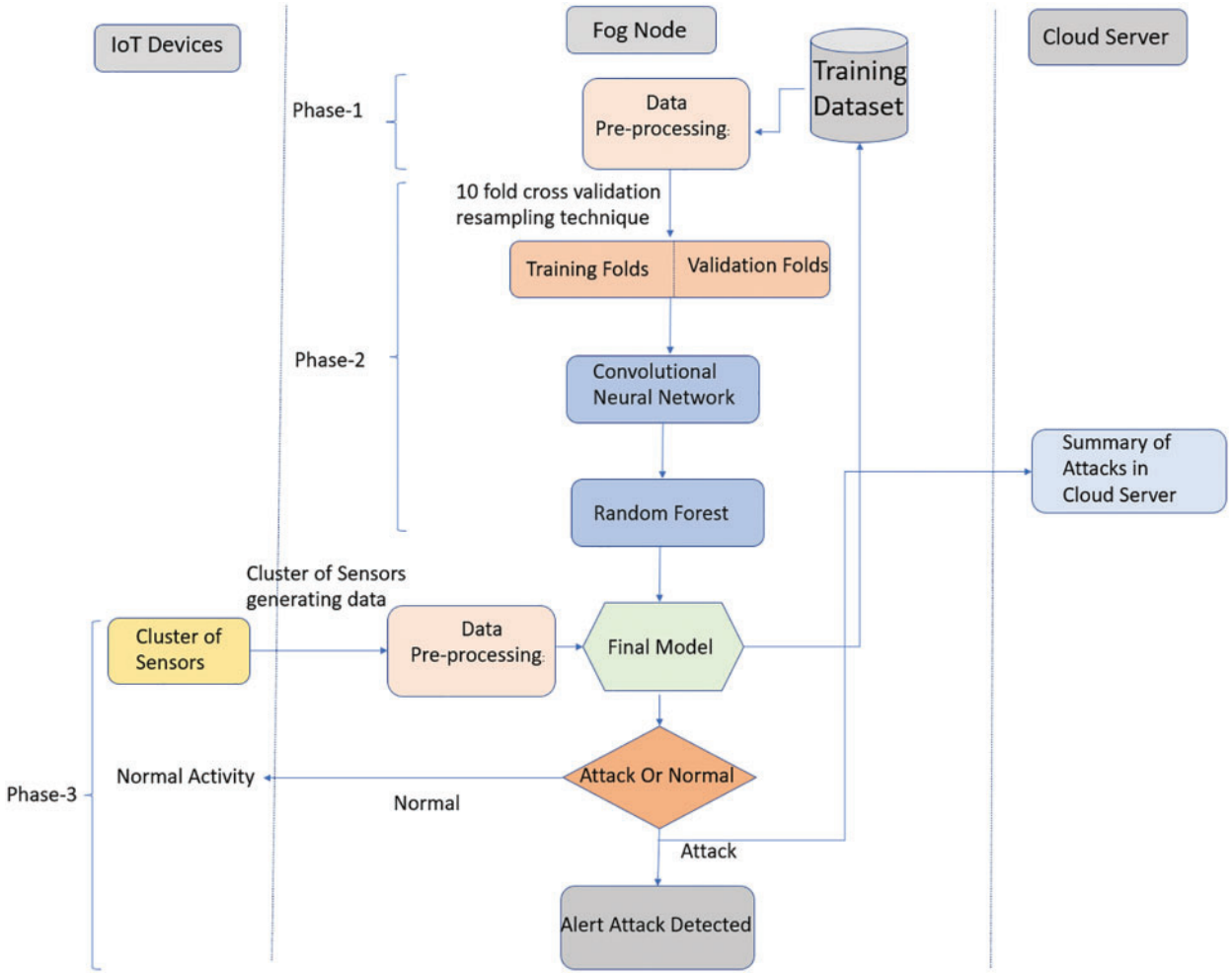

Figure 1: Working architecture of proposed distributed IDS model in fog computing

This research aims to use the characteristics of fog computing to design a distributed intrusion detection system to protect the security of the entire Internet of Things network and protect the entire Internet of Things system from various attacks.

The main contributions of our research are as follows:

1. We propose a distributed intrusion detection system based on convolutional neural network and random forest, using the characteristics of fog computing to protect the entire Internet of Things from attacks.
2. We put the proposed distributed intrusion detection system in the fog nodes of the entire Internet of Things to prove its effectiveness.
3. The performance of the distributed integrated intrusion detection system is evaluated through the UNSW-NB15 dataset. The biggest advantage of this dataset is that it has a variety of modern Internet of Things attacks and can make the system have modern protection functions.
4. In order to reduce the bias and variance, this study uses a 10-fold cross-validation resampling method.

The rest of this paper is organized as follows. In Section 2, the background and previous work related to this article are discussed. The Section 3 introduces the entire distributed and integrated design of the Internet of Things intrusion detection system, which is used to identify various Internet of Things attacks in the fog node. Section 4 presents the experimental results and system performance using the UNSW-NB15 dataset. Finally, the Section 5 summarizes the article.

2 Related Work

In recent years, with the popularity of machine learning algorithms and deep learning algorithms, many intrusion detection models have been proposed. Moustafa et al. [15] Proposed an intrusion detection model which uses the KDDCup99 and UNSW-NB15 datasets. The model uses decision tree (DT), expectation maximization (EM) clustering, artificial neural network (ANN), naïve Bayes (NB), and logistic regression (LR) methods. The accuracy of the model using UNSW-NB15 for DT is 85.56%, and the accuracy of using the KDDCup99 data set for ANN is 97.04%.

Pajouh et al. [16] proposed an intrusion detection system based on a two-layer classification model of machine learning in 2017. The model uses naïve Bayes (NB), certain factor voting version of KNN classifiers and also Linear Discriminant Analysis for dimension reduction for classification. All evaluation processes of this model use the NSL-KDD dataset. Pajouh et al. [17] proposed an intrusion detection system based on two-layer dimensionality reduction and two-layer classification modules in 2019 to detect user-to-Root and remote-to-local attacks. The model uses linear discriminant analysis in the dimensionality reduction module to reduce the dimensionality of high-dimensional datasets to low-dimensional datasets with fewer features, and uses Nave Bayes and Certainty Factor version of K-Nearest Neighbor in the classification module to identify various attacks. The model also uses the NSL-KDD dataset to evaluate and verify the effectiveness and superiority of the model, and the detection rate for binary classification reaches 84.66%.

Kumar et al. [18] proposed an intrusion detection system based on misuse that can identify five types of attacks in the network: Exploit, DOS, Probe, Generic, and Normal. The system uses the UNSW-NB15 dataset as an offline dataset, and designs a classification model. In addition, the system uses the RTNITP18 dataset as a testing dataset to evaluate the performance of the proposed model. Kumar et al. [9] also proposed a new type of unified intrusion detection system for IoT environment (UIDS) in 2019, which can prevent four types of attacks: exploit, DoS, probe and generic. In addition, it can detect the traffic category in the normal network. The system uses the UNSW-NB15 dataset as the benchmark dataset and is designed to detect the UIDS of malicious activities on the network.

In addition, it can be found that most of the current intrusion detection systems will be designed as a centralized computing architecture. This architecture will have low detection accuracy and high false alarm rate in the actual environment. Therefore, this paper proposes an intrusion detection system based on fog computing with distributed design. In this paper, the intrusion detection system is placed in the fog node, which greatly reduces the storage in the cloud server and the delay of the entire IoT system, and is used to prevent various malicious attacks.

3 Proposed Work

This section introduces the detailed design and specific architecture of the proposed distributed integrated intrusion detection system. Fig. 1 is the anomaly-based distributed intrusion detection system architecture in the entire Internet of Things system. This model puts an anomaly-based distributed intrusion detection system in the fog node. The IoT device sends the collected data to the

fog node. The fog node uses the intrusion detection system to determine whether the current network is under attack, and sends the information after the judgment to the cloud server. This is the entire distributed architecture. The whole model consists of three phases.

In the first phase, we need to preprocess the training data set, which includes feature mapping, imputing missing values, normalization and mutual information-based feature selection. We get the optimized feature set through these data pre-processing. In the second phase, we train the convolutional neural network with the optimized feature set by using the 10-fold cross-validation resampling method technology. This is the first layer of the model. The 10-fold cross-validation resampling technique is to randomly divide the data set into 10 parts, use 9 of them for training, and use the other for testing. This process can be iterated 10 times, and the testing dataset used in each iteration is different, which will make the classifier more accurate. Then this model use the trained result as the input of the random forest, and finally get the final classification. In the third stage, the data collected by the Internet of Things devices is collected through the fog node, and the collected raw data is processed by the same data preprocessing method before, and then the processed data is used as the input of the model to verify the effectiveness of the intrusion detection system under the entire physical network system. If the predicted result is normal traffic, it means that the current IoT system has not been maliciously attacked and the IoT devices can be allowed to perform normal activities. If the predicted result is abnormal traffic, the fog node will issue an alert to the cloud server, and define the IoT device that sends this abnormal traffic as an IoT botnet to prevent its operation. At the same time, logs are recorded in the cloud server to share attack information on the entire network. The cloud server maintains the global status of IoT devices and performs global management and control. Fig. 2 is the flowchart of the proposed intrusion detection system. We will introduce the above three stages in sequence in the following subsections.

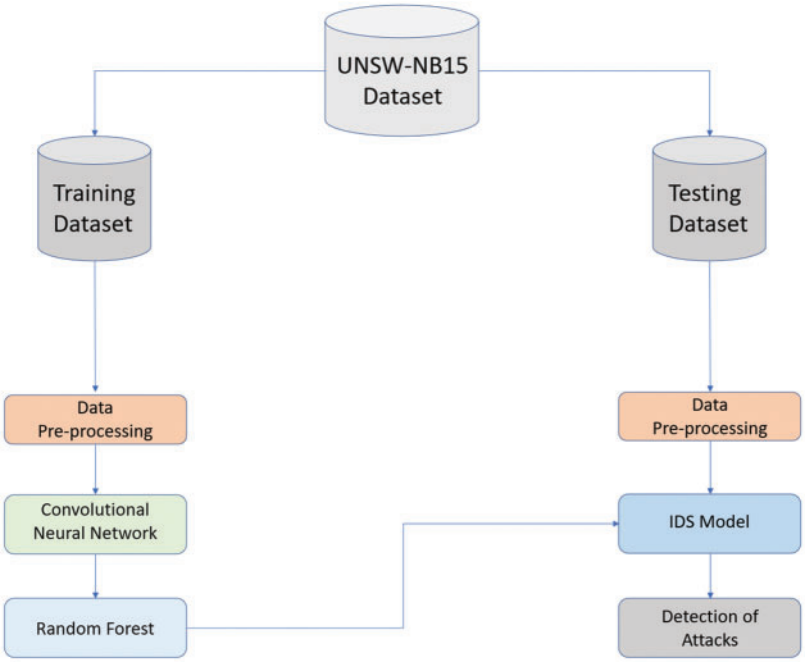

Figure 2: Flowchart of proposed distributed IDS model

3.1 Data Pre-Processing

Since the data sent by different IoT devices have different characteristics, it is very necessary to preprocess the data. Data pre-processing mainly includes feature mapping, imputing missing values, normalization and mutual information-based feature selection.

3.1.1 Feature Mapping

Since the data sent by IoT devices to the fog node contains various features, and we deal with digital features best, we need a mapping technology that converts various features into digital features. This paper combines Label-one-hot-encoding (LOHE) and one-hot encoding (OHE) as a mapping technology. LOHE technology maps each classification feature to a digital integer. OHE creates a new column for the type data, because OHE has only one bit valid at any time, that is to say, its value is only 0 or 1, and different types are stored in the vertical space. It turns out that this is very effective for the proposed intrusion detection system.

3.1.2 Imputing Missing Values

Since the traffic in the Internet of Things usually contains missing values, it is usually coded as blanks, NaN or other placeholders. In order to perform a unified meaningful analysis, we replace the missing type value with the unknown class, and replace the missing integer value with the mean value of the current feature.

3.1.3 Normalization

Since the traffic in the Internet of Things has different magnitude values, it needs to be standardized in this model. Standardization is the process of scaling a single sample to have a unit norm. The model uses StandardScaler standardization technology, which makes the mean value of the flow 0 and the standard deviation of 1. This technology eliminates any deviation of the traffic incoming to the fog node, and does not change its statistical properties. The standardized transformation function is Eq. (1).

$$s_k = \frac{v_i - \mu_k}{\sigma_k} \tag{1}$$

At this time s_k is the standardized score $k \in \{k1, k2, k3 \ldots, kn\}$. Among them, the corresponding characteristic value of the traffic in the Internet of Things is represented by v_i. The average value of traffic in the Internet of Things is represented by μ_k, which is calculated using Eq. (2). The standard deviation of the traffic in the Internet of Things is represented by σ_k, which is calculated using the formula of Eq. (3).

$$\mu_k = \frac{\sum_{i=1}^{N} v_i}{N} \tag{2}$$

$$\sigma_k = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(v_i - \mu_k)^2} \tag{3}$$

3.1.4 Feature Selection

Feature selection is a process of removing repetitive and irrelevant features. In this process, this paper uses mutual information-based feature selection technique [19]. This technology has a feature

and label class, mutual information is a measure of the degree of mutual dependence between the feature and the label class. This technique uses two given random features K1 and H2, these two random features serve as features and label classes, and use feature H2 to estimate the amount of information obtained about feature K1. The following is based on the mutual information function Eq. (4).

$$I(K1; H2) = \int_{K1} \int_{H2} p(K1, H2) \log \frac{p(K1, H2)}{p(K1)\, p(H2)} dH2\, dK1 \tag{4}$$

where $p(K1, H2)$ is the joint probability function of features K1 and H2, $p(K1)$ and $p(K2)$ are the respective edge density functions. Because the greater the mutual information between the feature and the tag class, the stronger the correlation between the feature and the tag, and the more likely this feature is to belong to this type of tag, so we maximize the mutual information between two random features.

3.2 Ensemble Learning Model

After data pre-processing, we will get the optimized feature set, and then in this section we will discuss in detail using the optimized feature set for two-layer classification training. The first layer is a convolutional neural network, and the second layer is a random forest method. The 10 fold cross validation resampling method is used when training the first layer.

In this model, the convolutional neural network is used as the first layer of learning and training model, and then the obtained output is used as the input of the random forest. After learning and training in random forest, an IDS that can adapt to fog computing is constructed to protect the security of the Internet of Things.

Many previous studies used machine learning algorithms when building intrusion detection systems, such as k-NN, XGBoost and NB. However, there is now a subset of machine learning: deep learning is developing very rapidly. Deep learning is a machine learning method that is used to build and simulate the neural network of the human brain for analysis and learning, and to imitate the mechanism of the human brain to interpret data. After the amount of data reaches a certain level, deep learning will have better performance than previous machine learning algorithms. Convolutional neural network is one of the most representative deep learning algorithms. It is a feed-forward neural network, and in our proposed model, the convolutional neural network can effectively classify various features, so we use Convolutional Neural Network in the first layer.

3.2.1 Convolutional Neural Network

Convolutional neural network is essentially a feed-forward deep artificial neural network, mainly used for vision-related applications. CNN has an input layer, one or more hidden layers that perform linear or non-linear transformations, and an output layer. The hidden layer of CNN is generally composed of a convolutional layer, a pooling layer, a batch normalization layer, an activation layer, and a fully connected layer [20]. In our proposed intrusion detection model, we first use the 10 fold cross validation resampling method to randomly divide the training set into 10 parts, nine of which are used as the training dataset, and the other one is used as the testing dataset. Then train the convolutional neural network on the nine training datasets, and then test the trained CNN model on the testing dataset. This process is iterated 10 times, and the data set used as the test set should be different each time, which ensures that the final CNN model will be more accurate.

3.2.2 Random Forest

Random forest is composed of many decision trees, and there is no correlation between different decision trees. When our model performs a classification task, when a new input sample enters, each decision tree in the forest will be judged and classified separately, and each decision tree will get its own classification result. In the end, the random forest will consider which of the classification results of the decision tree has the most classification as the final result. There are four steps to construct a random forest:

1. Random sampling, training decision tree
2. Randomly select features and do node splitting attributes
3. Repeat step 2 until it can no longer split
4. Establish a large number of decision trees to form a forest

In our model, the random forest uses the prediction results obtained by the convolutional neural network as a meta-classifier, and then the normal and attack existing in the training set are finally classified. Tab. 2 is that Algorithm 1 explains the steps involved in the entire model. Tab. 1 is an explanation of the symbols that need to be used in Algorithm 1.

Table 1: List of notation used in ensemble learning model

List of notations used	Meaning
F^n	Feature space
τ	Class label set
B	Base classifier
E	Ensemble classifier
δ	Number of training examples
$T = \{P_i, Y_i\}_{i=1}^{\delta}$ $(P_i \in F^n, Y_i \in \tau)$	Training dataset

Table 2: Algorithm 1: Steps of ensemble learning model

Algorithm 1: Steps of Ensemble Learning Model

Input: Training Dataset $T = \{P_i, Y_i\}_{i=1}^{\delta}$ $(P_i \in F^n, Y_i \in \tau)$
Output: An Ensemble Learning Classifier
Step 1: Training the first-level classifiers B
Construction of CNN;
$N = 50$;
$i = 1$;
Repeat
Step 1.1: using 10-fold cross validation resampling technique to generate training set $T = \{T_1, T_2, T_3 \ldots, T_{10}\}$
Step 1.2: Training the base-level classifiers
for $j \leftarrow 1$ to 10 **do**

(Continued)

Table 2: Continued

for $k \leftarrow 1$ to 10 **do**
Training the CNN model with T_k, $k \neq j$;
Testing the model with T_j;
End
End
$i = i+1$;
Until $i = N$
Step 2: Training the second-level classifiers E using Random Forest
Return E

3.3 Layered Architecture for Test Operation

The intrusion detection model proposed in this paper is based on different attack scenarios, because there are complex and different attack scenarios in the Internet of Things network, so we adopt a hierarchical structure. In the hierarchical structure, the integrated intrusion detection system uses the data transmitted by the Internet of Things devices for testing, and compares the incoming traffic with the current attack type. If it gets an alarm from the administrator, it means that it has been the attack categories are matched, and there is no need to perform the next step of comparison. If it does not match the current attack type, compare it with the next attack type. The comparison method is the same as the first attack type. If the match is not successful until the end, it means that the incoming traffic is normal traffic, so the IoT device can be allowed to perform regular activities. Experimental results prove that the system is persistent against various types of attacks, and detects malicious traffic in the Internet of Things.

4 Experimental Results and Discussion

This section performs performance analysis and results display of the proposed integrated intrusion detection model, and compares it with the existing intrusion detection model. The experiment was carried out using the Python programming language, PyTorch was used to build the CNN model, and the UNSW-NB15 dataset was used for training and testing. Finally, the performance of the model was tested.

4.1 Description of UNSW-NB15 Dataset

This model uses the UNSW-NB15 dataset [21] to train and test the model, and evaluate it. This dataset is considered a modern dataset. There are 9 kinds of attacks in this dataset, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. The dataset has a total of 49 features, a total of 175341 records in the training set, and a total of 82332 records in the test set (Tabs. 3 and 4).

Table 3: Distributions of attacks and normal instances in UNSW-NB15 training dataset

Division of class	Total occurrences	Percentage of class frequency
Analysis	2000	1.14

(Continued)

Table 3: Continued

Division of class	Total occurrences	Percentage of class frequency
Backdoor	1746	0.99
Normal	56000	31.93
Fuzzers	18184	10.37
Dos	12264	6.99
Reconnaissance	10491	5.98
Generic	40000	22.81
Exploits	33393	19.04
Worms	130	0.07
Shellcode	1133	0.64
Total	175341	100

Table 4: Distributions of attacks and normal instances in UNSW-NB15 testing dataset

Division of class	Total occurrences	Percentage of class frequency
Analysis	677	0.82
Backdoor	583	0.70
Normal	37000	44.93
Fuzzers	6062	7.39
Dos	4089	4.96
Reconnaissance	3496	4.24
Generic	18871	22.92
Exploits	11132	13.52
Worms	44	0.05
Shellcode	378	0.45
Total	82332	100

4.2 Description of Evaluation Metrics

We use different evaluation indicators to verify the proposed model, such as: Accuracy, detection rate, false alarm rate, precision.

In this section we need to use the following defined indicators:

l True Positive (TP)- Attack data that is correctly classified as an attack.
l False Positive (FP)- Normal data that is incorrectly classified as an attack.
l True Negative (TN)- Normal data that is correctly classified as normal.
l False Negative (FN)- Attack data that is incorrectly classified as normal.
a) Accuracy (AC) [22]: The accuracy measures the proportion of the total number of correct classifications Eq. (5).

$$AC = \frac{TP + TN}{FN + TP + FP + TN} \tag{5}$$

b) Precision (PR) [22]: The precision measures the number of correct classifications penalized by the number of incorrect classifications Eq. (6).

$$PR = \frac{TP}{TP + FP} \tag{6}$$

c) Detection rate (DR) [22]: The DR or recall (RC) measures the number of correct classifications penalized by the number of missed entries Eq. (7).

$$DR = \frac{TP}{TP + FP} \tag{7}$$

d) False alarm rate (FAR) [22]: The false alarm measures the proportion of benign events incorrectly classified as malicious Eq. (8).

$$FAR = \frac{FP}{FP + TN} \tag{8}$$

e) F1 score [22]: The F1-score measures the harmonic mean of precision and recall, which serves as a derived effectiveness measurement Eq. (9).

$$F1 = 2 \times \frac{RC \times PR}{RC + PR} \tag{9}$$

4.3 Evaluation and Discussion of Results

This model attempts to overcome the shortcomings of existing intrusion detection systems in AC, DR, PR, FAR. The system uses two different feature sets (20 and 30 feature sets) for evaluation. Tab. 5 shows the prediction results for all attack categories using 20 and 30 features. Backdoor attack performance results show that the effect is best in the case of 30 features. With 30 features, AC reaches 95.42%, PR reaches 28.54%, DR reaches 70.35%, and FAR reaches 2.35%. For Analysis, the attack performance results show that the best effect is achieved in the case of 30 feature sets, at which AC reaches 95.24%, PR 31.56%, DR 67.84% and FAR 2.53%. For Fuzzers, the results show that the effect is the best under the condition of 30 feature sets. When there are 30 features, AC reaches 77.53%, PR 33.26%, DR 58.36% and FAR 17.24%. For Shellcode, the attack performance results show that the attack performance is best in the case of 20 feature sets, at which AC, PR, DR and FAR reach 96.25%, 20.53%, 70.51% and 2.56% respectively. Reconnaissance attack performance results show that the effect is best in the case of 20 features. In the case of 20 features, AC reached 93.45%, PR reached 67.24%, DR reached 70.53%, and FAR reached 3.15%. Exploits attack performance results show that the effect is best in the case of 30 features. In the 30 features, AC reached 88.54%, PR reached 70.26%, DR reached 87.25%, and FAR reached 10.26%. For Dos attack, the attack performance results show that the best effect is achieved in the case of 30 feature sets. In the case of 30 feature sets, AC reaches 90.35%, PR 53.72%, DR 86.25% and FAR 7.25%. However, for Worms, the attack performance results show that the best effect is achieved in the case of 20 feature sets, at which AC reaches 99.88%, PR 61.32%, DR 35.32% and FAR 0.05%. Generic attack performance results show that the effect is best in the case of 30 features. In the case of 30 features, AC reaches 96.87%, PR reaches 92.34%, DR reaches 94.25%, and FAR reaches 3.14%. The results show (Figs. 3 and 4) that The model has better performance when using 30 features to detect different attacks than when using 20 feature sets.

Table 5: Prediction results for UNSW-NB15 dataset

Attacks	20 features					30 features				
	AC	PR	DR	F1	FAR	AC	PR	DR	F1	FAR
Backdoor	94.38	20.63	65.24	30.45	3.46	95.42	28.54	70.35	39.24	2.35
Analysis	94.24	20.42	65.25	31.65	3.25	95.24	31.56	67.84	43.15	2.53
Fuzzers	73.64	32.54	64.25	41.26	22.46	77.53	33.26	58.36	42.62	17.24
Shellcode	96.25	20.53	70.51	31.45	2.56	95.25	20.53	72.45	30.76	2.87
Reconna.	93.45	67.24	70.53	68.26	3.15	92.15	56.58	91.25	70.28	6.25
Exploits	81.26	58.72	81.32	67.32	16.25	88.54	70.26	87.25	77.25	10.26
Dos	84.23	32.45	51.46	40.26	10.46	90.35	53.72	86.25	65.32	7.25
Worms	99.88	61.23	35.32	42.25	0.05	99.86	46.97	54.25	50.27	0.10
Generic	96.25	91.25	95.52	93.26	3.28	96.87	92.34	94.25	93.24	3.14

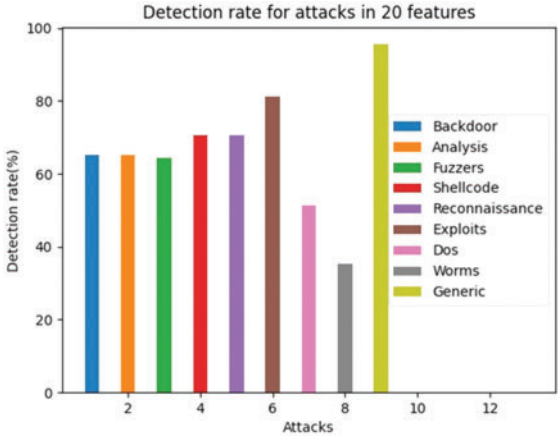

Figure 3: Detection rate for attacks in 20 features

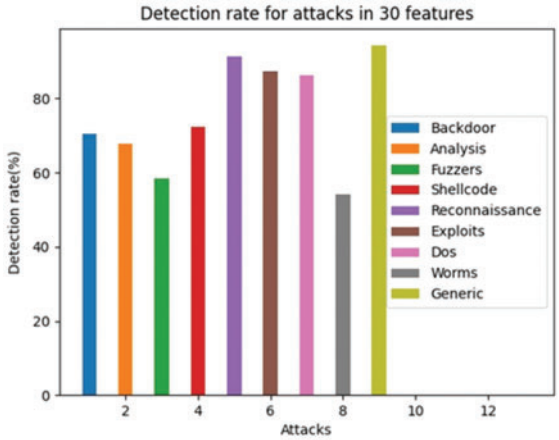

Figure 4: Detection rate for attacks in 30 features

4.4 Results Comparison with Existing IDS

Tab. 6 shows the comparison of various predictive indicators between our proposed model and the existing intrusion detection model when the UNSW-NB15 dataset is used. Compared with Kumar et al. [9,18], Papamartzivanos et al. [23], Ren et al. [24], and Khan et al. [25], our proposed model can be Effectively improve the detection rate, for example: backdoor 70.35% DR, analysis 67.84% DR, Dos 86.25% DR.

Table 6: Detection rate (%) comparison with other existing models for UNSW-NB15 dataset

Attack	Proposed (30)	UIDS (Kumar et al. [9])	RIDS (Kumar et al. [18])	DENDRON (Papamartzivanos et al. [23])	DO_IDS (Ren et al. [24])	TSDL(Khan et al. [25])
Backdoor	70.35	-	-	67.32	40.30	0.00
Analysis	67.84	-	-	20.45	6.10	16.35
Fuzzers	58.36	-	-	64.42	38.10	85.63
Shellcode	72.45	-	-	36.39	78.00	18.79
Reconna.	91.25	79.42	71.7	46.04	82.00	71.75
Exploits	87.25	85.69	54.64	76.22	66.30	92.40
Dos	86.25	29.46	5.0	14.29	46.10	0.48
Worms	54.25	-	-	18.37	79.50	0.00
Generic	94.25	98.58	96.72	81.37	96.90	97.76

5 Conclusion

Aiming at the various limitations of the existing centralized intrusion detection system, this paper proposes an intrusion detection system based on distributed integrated design. In this paper, the distributed integrated intrusion detection system is placed in the fog node in the Internet of Things network, and the characteristics of fog computing are used to protect the entire Internet of Things network. The intrusion detection model has three stages in total. In the first stage, we need to preprocess the original data, including feature mapping and feature selection, to get the optimized feature set. In the second stage, we need to classify, we use a two-layer model of CNN combined with random forest. The first is to use the 10-fold cross-validation resampling technique to select the training set and the test set, and then to train the model of CNN combined with random forest. In the third stage, we transfer the data collected from the IoT device to the fog node, and our trained model will test the data on the fog node. Normal traffic allows activity, and abnormal traffic restricts its activity. Finally, we use the UNSW-NB15 data set to evaluate the performance of the model. The experimental results show that our proposed model has a higher detection rate than most existing centralized intrusion detection models. In the future, we will make further improvements to the model. We plan to use more advanced technologies to extend the model and strive to be able to identify more modern attacks.

Funding Statement: This work is supported in part by the Beijing Natural Science Foundation (No. 4212015), Natural Science Foundation of China (No. 61801008), China Ministry of Education-China Mobile Scientific Research Foundation (No. MCM20200102), China Postdoctoral Science

Foundation (No. 2020M670074), Beijing Municipal Commission of Education Foundation (No. KM201910005025).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal *et al.,* "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019.
[2] B. Sinha and R. Dhanalakshmi, "Recent advancements and challenges of internet of things in smart agriculture: A survey," *Future Generation Computer Systems*, vol. 126, no. 4, pp. 169–184, 2021.
[3] Q. Tang and F. Du, "Internet of things security: Principles and practice," 2021.
[4] M. F. Elrawy, A. I. Awad and H. Hamed, "Intrusion detection systems for IoT-based smart environments: A survey," *Journal of Cloud Computing*, vol. 7, no. 1, pp. 1–20, 2018.
[5] S. Venticinque and A. Amato, "A methodology for deployment of IoT application in fog," *Journal of Ambient Intelligence & Humanized Computing*, vol. 10, pp. 1955–1976, 2018.
[6] A. Alrawais, A. Alhothaily, C. Hu and X. Cheng, "Fog computing for the internet of things: Security and privacy issues," *IEEE Internet Computing*, vol. 21, no. 2, pp. 34–42, 2017.
[7] K. Costa, J. Papa, C. Lisboa, R. Munoz and A. D. Albuquerque, "Internet of things: A survey on machine learning-based intrusion detection approaches," *Computer Networks*, vol. 151, no. 14, pp. 147–157, 2019.
[8] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 20, pp. 22, 2019.
[9] V. Kumar, A. K. Das and D. Sinha, "UIDS: A unified intrusion detection system for IoT environment," *Evolutionary Intelligence*, vol. 14, no. 1, pp. 47–59, 2021.
[10] N. Moustafa, "Designing an online and reliable statistical anomaly detection framework for dealing with large high-speed network traffic," 2017.
[11] G. Creech and J. Hu, "Generation of a new IDS test dataset: Time to retire the KDD collection," in *2013 IEEE Wireless Communications and Networking Conf. (WCNC)*, Shanghai, China, pp. 4487–4492, 2013.
[12] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symp. on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, Canada, pp. 1–6, 2009.
[13] B. Sangster, T. J. O'Connor, T. Cook, R. Fanelli, E. Dean *et al.,* "Toward instrumenting network warfare competitions to generate labeled datasets," *CSET*, vol. 9, pp. 1–6, 2009.
[14] I. Perona, I. Gurrutxaga, O. Arbelaitz, J. I. Martin, J. Muguerza *et al.,* "Service-independent payload analysis to improve intrusion detection in network traffic," in *Proc. of the 7th Australasian Data Mining Conf.*, Glenelg, South Australia, vol. 87, pp. 171–178, 2008.
[15] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Systems Security*, vol. 25, no. 1–3, pp. 18–31, 2016.
[16] H. H. Pajouh, G. H. Dastghaibyfard and S. Hashemi, "Two-tier network anomaly detection model: A machine learning approach," *Journal of Intelligent Information Systems*, vol. 48, no. 1, pp. 1–14, 2015.
[17] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha and K. R. Choo, "A Two-layer dimension reduction and Two-tier classification model for anomaly-based intrusion detection in IoT backbone networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 2, pp. 314–323, 2019.
[18] V. Kumar, D. Sinha, A. Das, S. Pandey and R. Goswami, "An integrated rule based intrusion detection system: Analysis on UNSW-NB15 data set and the real time online dataset," *Cluster Computing*, vol. 23, no. 2, pp. 1397–1418, 2020.

[19] A. Sprm, A. Pkrm, A. Pm, A. Sk, A. Trg *et al.,* "An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture," *Computer Communications*, vol. 160, pp. 139–149, 2020.
[20] A. Khan, A. Sohail, U. Zahoora and A. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020.
[21] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conf. (MilCIS)*, Canberra, ACT, Australia, pp. 1–6, 2015.
[22] M. C. Belavagi and B. Muniyal, "Performance evaluation of supervised machine learning algorithms for intrusion detection," *Procedia Computer Science*, vol. 89, pp. 117–123, 2016.
[23] D. Papamartzivanos, F. G. Mármol and G. Kambourakis, "Dendron: Genetic trees driven rule induction for network intrusion detection systems," *Future Generation Computer Systems*, vol. 79, pp. 558–574, 2018.
[24] J. D. Ren, J. W. Guo, Q. Wang, Y. Huang and J. J. Hu, "Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithms," *Security and Communication Networks*, vol. 2019, no. 1, pp. 1–11, 2019.
[25] F. A. Khan, A. Gumaei, A. Derhab and A. Hussain, "A novel Two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30373–30385, 2019.