Journal of Quantum Computing
DOI:10.32604/jqc.2021.025373
Article

Tech Science Press

Intrusion Detection Method of Internet of Things Based on Multi GBDT Feature Dimensionality Reduction and Hierarchical Traffic Detection

Taifeng Pan*

Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China

*Corresponding Author: Taifeng Pan. Email: pantaifeng@bupt.edu.cn

Received: 02 October 2021; Accepted: 28 November 2021

Abstract: The rapid development of Internet of Things (IoT) technology has brought great convenience to people's life. However, the security protection capability of IoT is weak and vulnerable. Therefore, more protection needs to be done for the security of IoT. The paper proposes an intrusion detection method for IoT based on multi GBDT feature reduction and hierarchical traffic detection model. Firstly, GBDT is used to filter the features of IoT traffic data sets BoT-IoT and UNSW-NB15 to reduce the traffic feature dimension. At the same time, in order to improve the reliability of feature filtering, this paper constructs multiple GBDT models to filter the features of multiple sub data sets, and comprehensively evaluates the filtered features to find out the best alternative features. Then, two neural networks are trained with the two data sets after dimensionality reduction, and the traffic will be detected with the trained neural network. In order to improve the efficiency of traffic detection, this paper proposes a hierarchical traffic detection model, which can reduce the computational cost and time cost of detection process. Experiments show that the multi GBDT dimensionality reduction method can obtain better features than the traditional PCA dimensionality reduction method. Besides, the use of dual data sets improves the comprehensiveness of the IoT intrusion detection system, which can detect more types of attacks, and the hierarchical traffic model improves the detection efficiency of the system.

Keywords: IoT security; network traffic analysis; attack detection; machine learning

1 Introduction

The rapid development of the IoT has brought great convenience to people's life. However, with the increase of the IoT systems and equipment, the security problem of the IoT is becoming more and more serious and urgent. Due to the complexity and diversity of IoT attacks, the traditional intrusion detection scheme based on rule detection has been difficult to meet the current needs. Therefore, more and more researchers have turned their attention to machine learning.

Jan et al. [1] developed a lightweight attack detection strategy utilizing a supervised machine learning-based support vector machine (SVM) to detect an adversary attempting to inject unnecessary data into the IoT network. Alazzam et al. [2] proposed a wrapper feature selection algorithm for IDS. This algorithm used the pigeon inspired optimizer to utilize the selection process. Ravi et al. [3] proposed a new SDRK machine learning (ML) algorithm to detect intrusion. Lv et al. [4] proposed a novel accurate and effective misuse intrusion detection system that relies on specific attack signatures to distinguish between normal and malicious activities to detect various attacks based on an extreme learning machine with a hybrid kernel function (HKELM). Hassan et al. [5] proposed a hybrid deep learning model to efficiently detect network

intrusions based on a convolutional neural network (CNN) and a weight-dropped, long short-term memory (WDLSTM) network. Zhang et al. [6] proposed an intrusion detection model based on improved genetic algorithm (GA) and deep belief network (DBN). Yang et al. [7] put forward the LM-BP neural network model. The LM-BP neural network model was applied to an intrusion detection system, and the intrusion detection flow under LM-BP algorithm was given. Zarai et al. [8] proposed an intrusion detection system based on deep neural network and short-term memory artificial neural network. Li [9] proposed a malicious attack detection method for IoT based on clustering and classification. Shen [10] proposed an attack detection model based on DT-DNN, and implemented a lightweight attack detection system working at the transport layer. Han [11] designed a lightweight IoT traffic detection model Page-Net. The model can reasonably lay out network parameters according to the distribution characteristics of traffic characteristics, and achieve high detection accuracy with a small number of parameter scales, which is more suitable for deployment in edge environments. Jin [12] proposed an abnormal flow detection technology based on the mixed dimensions of time and space and based on the sliding window, which can improve the efficiency and accuracy of abnormal traffic detection, and has lower computational overhead. Chen [13] proposed a collaborative anomaly detection framework based on Internet of Things and studied the anomaly detection algorithm based on image.

2 Data Processing

In this paper, machine learning technology is used to detect the traffic of the Internet of Things. The training of machine learning depends on the appropriate data set, so the first step is to find the appropriate data set as the data of model training. After comparison, we finally chose BoT-IoT and UNSW-NB15 as data set. BoT-IoT data set simulates the attack data collected in the IoT environment including 4 attack categories. UNSW-NB15 is not a data set specifically for IoT traffic, but it contains modern attack type data, which is more in line with the characteristics of the display scene and has rich attack types, which can just make up for the lack of BoT-IoT attack types.

2.1 Data Balance and Encoding

After selecting the dataset, we need to process the dataset. The first step is to solve the problem of uneven data distribution. We randomly sample the samples which accounts for a large proportion, and supplement the samples which accounts for small proportion with SMOTE algorithm. Finally, we get a balanced data set. The data after balance processing is shown in Table 1 and Table 2.

Table 1: BoT-IoT sampled data

Category	Number
Normal	4750
Reconnaissance	2000
DoS	3485
DDoS	2988
Theft	1587
Total	14180

Table 2: UNSW-NB15 sampled data

Category	Number
Analysis	2000
Backdoor	2000
DoS	2000
Exploits	2000
Fuzzers	2000

Generic	2000
Normal	18000
Reconnaissance	2000
Shellcode	1511
Worms	674
Total	34185

There are a large number of discrete features in the traffic data set, such as protocol and state in the session. We need to convert these discrete features into a form that is easy to use by the machine learning algorithm, so we need to encode these discrete features as one-hot. One-hot coding is the representation of classification variables as binary vectors. This first requires mapping classification values to integer values. Then, each integer value is represented as a binary vector, which is zero except for the index of the integer, and it is marked as one.

2.2 Feature Dimensionality Reduction

Original dataset usually provides multiple features and we classify the category of samples according to these features. However, the features provided in the samples are not all useful. Many features do not play a role or even play a negative role in the classification of the samples, and too many features will increase the complexity of the classifier, resulting in longer model training and testing time. Therefore, it is necessary to reduce the dimension of the features before formally training the classification model. At present, the most commonly used feature dimensionality reduction method is PCA. The basic idea of PCA is to find the main axis direction of data, and form a new coordinate system by the main axis. The dimension here can be lower than the original dimension, and then the data is projected from the original coordinate system to the new coordinate system. This projection process is the process of dimension reduction. PCA has a good dimensionality reduction effect in many scenarios, but PCA only considers the data correlation between features and does not consider the role of labels in the dimensionality reduction process. Therefore, some information loss may be caused in the dimensionality reduction process, which may affect the training of classification model. In addition, the features after PCA dimensionality reduction have no practical meaning, and it is difficult for the real feature collector to directly collect these features. Therefore, it is necessary to carry out another PCA operation on the collected features in the operation stage of the flow detection system before they are sent to the classification model for discrimination, which increases the complexity and calculation of the system.

Therefore, this paper will use GBDT for feature screening to realize feature dimensionality reduction. GBDT is one of the boosting ensemble learning methods. It can be used for both classification and regression. It is composed of multiple decision trees. In each step of GBDT algorithm, a decision tree is used to fit the residual of the current learner to obtain a new weak learner. Combining the decision trees of each step, we get a strong learner. Assuming that the sample has n features and the GBDT model has M decision trees, the importance of the $feature_i$ of the sample in the GBDT model is calculated as formula (1).

$$J_i = \frac{1}{M}\sum_{m=1}^{M} J_i(T_m) \tag{1}$$

In formula (1), J_i is the importance of the $feature_i$ in the global GBDT, and $J_i(T_m)$ represents the importance of the $feature_i$ in the $decision - tree_m$. $J_i(T_m)$ is determined by the change of impurity of the decision tree during node splitting.

There are two ways to express the impurity in the decision tree: Gini coefficient and information entropy. Taking Gini coefficient as an example, the Gini coefficient of a node in the decision tree is calculated as formula (2).

$$GINI_{node} = \sum_{k=1}^{K} p_k(1 - p_k) \tag{2}$$

In formula (2), K is the category quantity, p_k is the proportion of class k samples in node. When the node splitting is based on the $feature_i$, the change of impurity is calculated as formula (3).

$$\Delta GINI_{node}^{i} = GINI_{node} - GINI_{left} - GINI_{right} \quad (3)$$

In formula (3), $left$ and $right$ represent two new nodes after node splitting. Seeking the change of impurity is a greedy process, so the $feature_i$ selected during the node splitting must maximize the change of impurity in this splitting. After X rounds of splitting, the construction process of the decision tree is completed. At this time, the importance of a $feature_i$ in the decision tree T_m is evaluated as formula (4).

$$J_i(T_m) = \frac{\Delta GINI_{node}^{i}}{\sum_{j=1}^{X} \Delta GINI_{node}^{j}} \quad (4)$$

In order to further strengthen the reliability of feature reduction and reduce errors, this paper integrates the bagging idea of Random Forest when using GBDT for feature filtering, that is, divide multiple groups of samples, construct multiple GBDT models, use these GBDT models to filter features on the sub data sets respectively, and finally comprehensively choose the features filtered by multiple GBDT. We call the GBDT dimension reduction method with bagging idea as multiple GBDT dimensionality reduction method.

Let the feature dimension of the data set before One -Hot coding be n_{origin}, the feature dimension after coding is n_{onehot}, $n_{onehot} > n_{origin}$. The feature dimension reduction process of IoT traffic in this paper is as follows:

(1) Encode the original data set with One-Hot coding method. After encoding, $feature_i$ in original data set becomes $\{F_{ik} | k = 1,2,3, \dots, K_i\}$. For convenience of presentation, write F_{ik} as F_j. Then the original feature space $\{feature_i | i = 1,2, \dots, n_{origin}\}$ is mapped to the new feature space $\{F_j | j = 1,2, \dots, n_{onehot}\}$.

(2) Divide the data set encoded by One-Hot method into V groups, then we get V groups of sub data sets, $\{SubDataSet_v \mid v = 1,2, \dots, V\}$.

(3) Score the samples in $SubDataSet_v$ according to its importance with GBDT model, The specific scoring process is shown in formulas (1)~(4). The score of $F_j^{(v)}$ is $J_j^{(v)}$, $J_j^{(v)}$ represents the importance score of the j-th feature in $SubDataSet_v$ in feature space $\{F_j | j = 1,2, \dots, n_{onehot}\}$.

(4) Repeat Step (3) for V times to obtain the importance score of each feature in all sub data sets, $\left\{\{J_j^{(v)} \;\middle|\; j = 1,2, \dots n_{onehot}\} \mid v = 1, 2, \dots, V\right\}$.

(5) Calculate the final importance of each feature, $\overline{J_J} = \frac{J_j^{(v)}}{\sum_{v=1}^{V} J_j^{(v)}}$, Where $\overline{J_J}$ represents the comprehensive performance of the F_j in the V groups of feature set.

(6) Aggregate $\{F_j | j = 1,2, \dots, n_{onehot}\}$ according previous One-Hot method, then the feature dimension becomes n_{origin}. The new importance score of $feature_i$ after aggregation is $\overline{JJ_\iota} = \sum_{k=1}^{K_i} \overline{J_J} = \sum_{k=1}^{K_i} \overline{J_{\iota k}}$.

(7) Rank $\{feature_i | i = 1,2, \dots, n_{origin}\}$ according the importance score $\{\overline{JJ_\iota} | i = 1,2, \dots, n_{origin}\}$, and select S features with the highest score to form the feature set $\{feature_i \mid i = 1, 2, \dots, S\}$, these S features are the features after final dimensionality reduction selection.

3 Hierarchical Detection Model

3.1 Model Construction

In order to improve the comprehensiveness of the IoT traffic detection model, this paper selects two data sets for training, and finally obtains two neural network models FNN1 and FNN2, which are deployed in the intrusion detection system to detect the traffic in real time. However, the two models mean that the traffic needs to be detected twice, which often increases the delay of traffic detection. For some edge computing devices with limited resources, twice detection means double the amount of calculation, which will also bring great computing pressure to these edge computing devices. Therefore, the dual network

model needs to be improved to improve the detection efficiency. Therefore, a hierarchical detection model is constructed. Fig. 1 shows the structure of dual network parallel detection model and Fig. 2 shows the structure of hierarchical detection model.

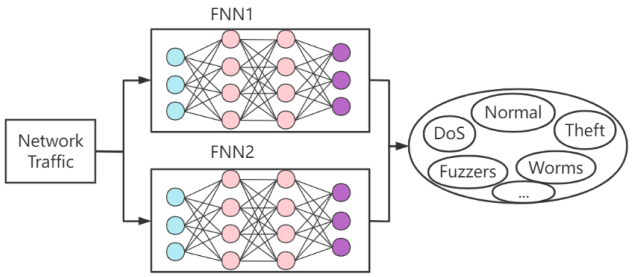

Figure 1: Dual network parallel detection model

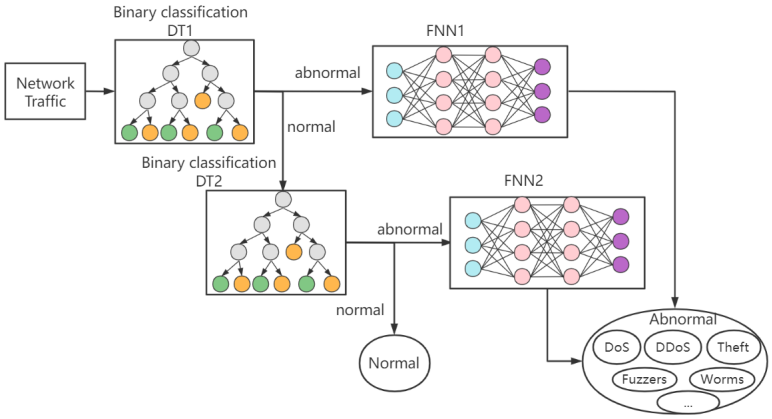

Figure 2: Hierarchical detection model

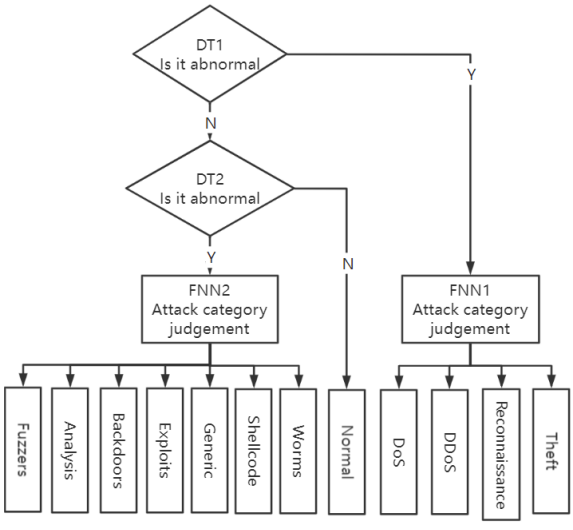

Figure 3: Hierarchical detection process

The hierarchical detection model consists of two binary decision trees and two fully connected neural networks. Binary classification DT1 is a decision tree model trained with BoT-IoT data set, and binary classification DT2 is a decision tree trained with UNSW-NB15 data set. FNN1 is a fully connected neural network trained with BoT-IoT data set, and FNN2 is a fully connected neural network trained with UNSW-NB15 data set. When the detection is started, the traffic is detected by the binary classification DT1 to

determine whether the traffic is normal or abnormal. If it is abnormal, FNN1 is activated, and FNN1 determines the specific attack type and gives an alarm. If DT1 determines that the traffic is normal, DT2 will be activated, and DT2 will judge whether it is normal. If it is normal, the output is normal. If it is abnormal, FNN2 will be activated, and FNN2 will determines the specific attack type. The classification process is shown in Fig. 3.

3.2 Performance Analysis of Hierarchical Detection Model

The performance of decision tree in complex classification scenarios is inferior to that of deep learning neural network, but it has a good performance in simple binary classification problems. Because the structure of binary classification decision tree is simple, it is better than complex neural network in detection speed and computation. This paper tests the time cost of decision tree and neural network with the same classification task on ARM platform and x86 platform respectively. The results show that the time cost of neural network is 4~5 times that of ARM. The specific experimental data will be shown in detail in Section 4.

In order to better evaluate the efficiency of the model, we set some variables and they are shown in Table 3.

Table 3: Model performance evaluation variables

Variables	Description
P_{d1}	Probability of abnormal traffic detected by DT1
T_{d1}	Time cost for DT1 to detect a single record
C_{d1}	Computational cost for DT1 to detect a single record
T_{f1}	Time cost for FNN1 to detect a single record
C_{f1}	Computational cost for FNN1 to detect a single record
P_{d2}	Probability of abnormal traffic detected by DT2
T_{d2}	Time cost for DT2 to detect a single record
C_{d2}	Computational cost for DT2 to detect a single record
T_{f2}	Time cost for FNN2 to detect a single record
C_{f2}	Computational cost for FNN2 to detect a single record
$\overline{T_\alpha}$	Time cost for dual FNN parallel model to detect a single record
$\overline{C_a}$	Computational cost for dual FNN parallel model to detect a single record
$\overline{T_\beta}$	Time cost for hierarchical model to detect a single record
$\overline{C_\beta}$	Computational cost for hierarchical model to detect a single record

Since it is difficult to directly evaluate computational cost, we equivalent the proportional relationship of computational cost to the proportion of time cost, that is, when the time cost of FNN is k times that of DT, it is considered that the computational cost of FNN is k times that of DT.

$$\frac{T_f}{T_d} = \frac{C_f}{C_d} = k \tag{5}$$

Then, the average time of detecting a single record in the dual network parallel detection mode shown in Fig. 1 is

$$\overline{T_\alpha} = \max(T_{f1}, T_{f2}) = \max(kT_{d1}, kT_{d2}) = k * \max(T_{d1}, T_{d2}) \tag{6}$$

Computational cost is

$$\overline{C_a} = C_{f1} + C_{f2} = kC_{d1} + \mathrm{k}C_{d2} = k(C_{d1} + C_{d2}) \tag{7}$$

The average time of detecting a single record in the hierarchical detection model shown in Fig. 2 is

$$\overline{T_\beta} = T_{d1} + P_{d1} * T_{f1} + (1 - P_{d1}) * [T_{d2} + P_{d2} * T_{f2}] \tag{8}$$

Computational cost is

$$\overline{C_\beta} = C_{d1} + P_{d1} * C_{f1} + (1 - P_{d1}) * \left[C_{d2} + P_{d2} * C_{f2}\right] \quad (9)$$

Experimental data show that T_f is about 3.7~5.2 times of T_d, we take the mean, 4.5 times, that is, $T_{f1} = 4.5T_{d1}$, $T_{f2} = 4.5T_{d2}$. Through formula (5), we can also get $C_{f1} = 4.5C_{d1}$, $C_{f2} = 4.5C_{d2}$. In the actual network environment, the proportion of normal traffic is much larger than that of abnormal traffic, so the probability of decision tree judging as normal traffic is very high. We set the proportion of abnormal flow in the actual environment as 5%, then $P_{d1} = 0.05$, $P_{d2} = 0.05$, so Table 4 can be obtained from formulas (6)–(9).

Table 4: Comparison of time cost and computational cost of the two models

$\overline{T_\alpha}$	$4.5 * \max(T_{d1}, T_{d2})$
$\overline{C_a}$	$4.5 * (C_{d1} + C_{d2})$
$\overline{T_\beta}$	$1.225T_{d1} + 1.164T_{d2}$
$\overline{C_\beta}$	$1.225C_{d1} + 1.164C_{d2}$

It can be seen from Table 4 that the hierarchical traffic detection model is superior to the dual network parallel detection in terms of time cost and computation cost.

4 Experimental Simulation

The original BoT-IoT dataset has 43 features and UNSW-NB15 has 47 features. The BoT-IoT and UNSW-NB15 data sets are processed by multiple GBDT dimensionality reduction method. Each data set retains 19 features, and the filtered features are shown in Table 5.

Table 5: Characteristics of GBDT after dimensionality reduction

Features after BoT-IoT filtering	Features after UNSW-NB15 filtering
flgs_number	service
bytes	sttl
sum	dttl
proto	state
max	proto
TnP_PDstIP	ct_dst_sport_ltm
dur	ct_dst_src_ltm
dbytes	ct_srv_dst
mean	sbytes
rate	tcprtt
TnP_Per_Dport	dbytes
sbytes	ct_state_ttl
N_IN_Conn_P_DstIP	smean
TnBPSrcIP	dmean
TnBPDstIP	sloss
TnP_PerProto	ct_flw_http_mthd
AR_P_Proto_P_DstIP	ct_src_ltm
srate	ct_src_dport_ltm
seq	dloss

In order to compare the dimensionality reduction effect of PCA and GBDT, the accuracy of PCA-DT, PCA-FNN, GBDT-DT and GBDT-FNN were tested on two data sets. The PCA dimension reduction retains 0.95 information, and the parameters of FNN are shown in Table 6. Fig. 4 shows the accuracy of different

models on BoT-IoT data set and Fig. 5 shows the accuracy of different models on UNSW-NB15 data set.

Table 6: FNN parameters

	FNN1(BoT-IoT)	FNN2(UNSW-NB15)
Number of hidden layers	3	3
Number of neurons in the first hidden layer	20	50
Number of neurons in the second hidden layer	20	20
Number of neurons in the second hidden layer	10	10
Optimizer	AdamW	AdamW
Activation function	gelu	gelu
Standardized method	LayerNorm	LayerNorm

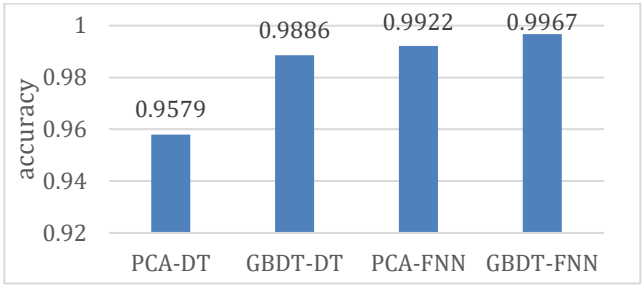

Figure 4: Comparison of accuracy of different models on BoT-IoT data set

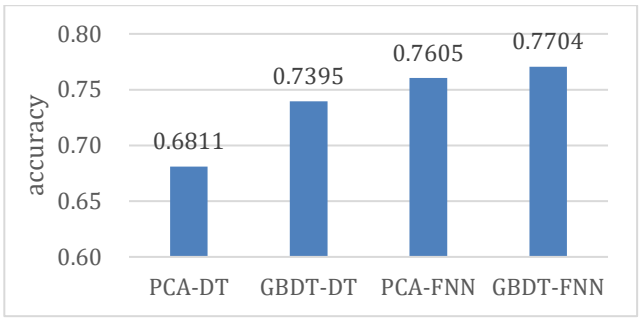

Figure 5: Comparison of accuracy of different models on UNSW-NB15 data set

Since UNSW-NB15 contains 10 types of attacks, and the display of recall and precision is very complex, this paper only shows the recall and precision on the BoT-IoT data set. Fig. 6 shows the recall rates of different models on BoT-IoT dataset and Fig. 7 shows the precision of different models on BoT-IoT data set.

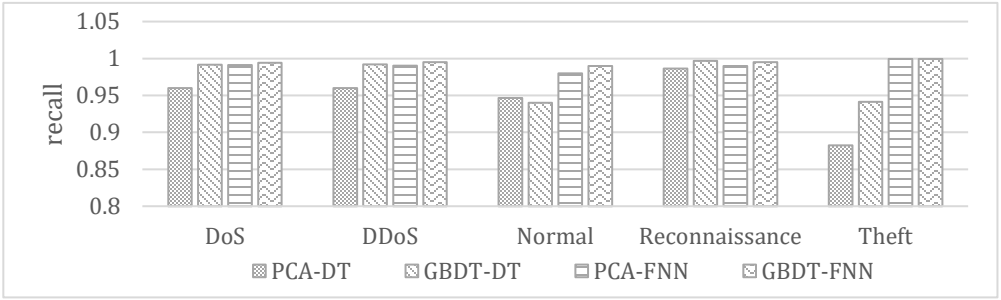

Figure 6: Comparison of recall rates of different models on BoT-IoT dataset

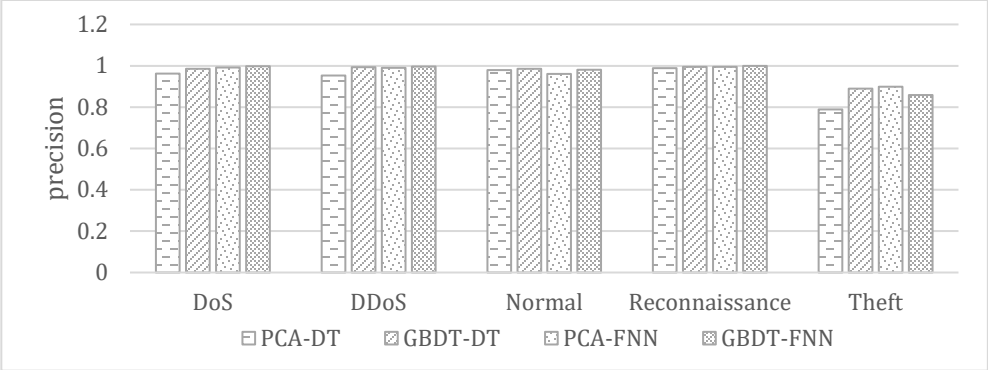

Figure 7: Comparison of precision of different models on BoT-IoT data set

In order to prove the rationality of the hierarchical detection model, we tested the DT binary classification accuracy, DT multi classification accuracy, FNN binary classification accuracy and FNN multi classification accuracy on the BoT-IoT data set and make a comparison.

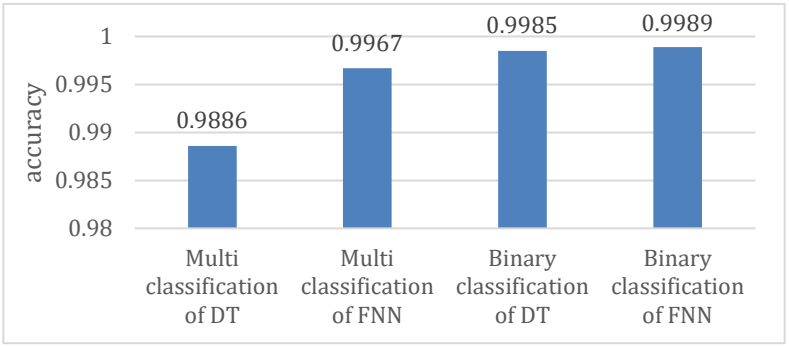

Figure 8: Comparison of DT and FNN classification on BoT-IoT dataset

It can be seen from Fig. 8 that FNN performs better than DT in the problem of multiple classification of abnormal traffic, but the difference between DT and FNN in the binary classification of normal traffic and abnormal traffic is very small. Therefore, taking DT as the binary classifier is reasonable.

In order to verify the detection efficiency of hierarchical detection model, the time cost of detecting a sample of binary classification decision tree and FNN model is tested on ARM platform and x86 platform. The experimental results are shown in Fig. 9 and Fig. 10.

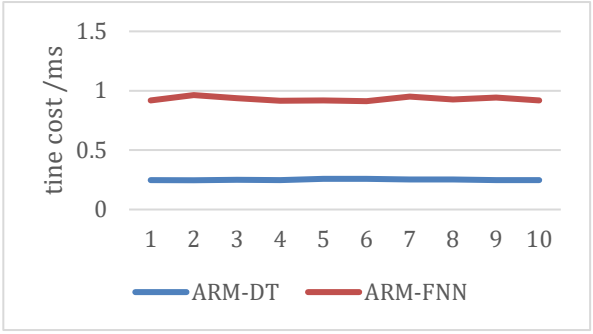

Figure 9: Time cost comparison between DT and FNN on ARM platform

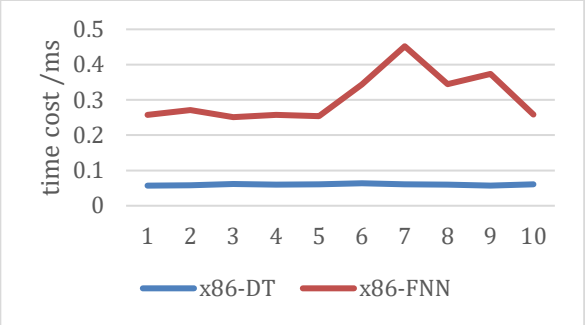

Figure 10: Time cost comparison between DT and FNN on x86 platform

As can be seen from Table 7, the time cost of FNN on ARM platform is 3.71 times that of DT, and that of FNN on x86 platform is 5.15 times that of DT.

Table: 7 Time cost of DT and FNN on two platforms

	ARM-DT	ARM-FNN	x86-DT	x86-FNN
Average time cost/ms	0.25060	0.93073	0.05988	0.30638

5 Summary

In order to deal with the attack of malicious IoT traffic, the paper proposes an IoT intrusion detection scheme based on multi GBDT feature dimensionality reduction and hierarchical traffic detection model. The detection scheme first uses the multiple GBDT model to reduce the dimension of two network traffic data sets, and then trains the dual network model with the processed two data sets. In order to improve the efficiency of traffic detection, a hierarchical detection model is proposed. The model is composed of two binary decision trees and two fully connected networks. The hierarchical detection model takes into account the characteristics of real network traffic, and combines the advantages of decision tree and neural network, it can improve the detection efficiency when detecting as many attack categories as possible.

Acknowledgement: We are grateful to the peoples for the support and encouragement.

Funding Statement: The author received no specific funding for this study.

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

References

[1] S. U. Jan, S. Ahmed and V. Shakhov, "Toward a lightweight intrusion detection system for the Internet of Things," *IEEE Access*, vol. 7, no. 1, pp. 42450–42471, 2019.

[2] H. Alazzam, A. Sharieh and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer," *Expert Systems with Applications*, vol. 148, no. 1, pp. 113249–113255, 2020.

[3] N. Ravi and S. M. Shalinie, "Semisupervised-learning-based security to detect and mitigate intrusions in IoT network," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 11041–11052, 2020.

[4] L. Lv, W. Wang and Z. Zhang, "A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine," *Knowledge-Based Systems*, vol.195, 105648, 2020.

[5] M. M. Hassan, A. Gumaei and A. Alsanad, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Information Sciences*, vol. 513, no. 1, pp. 386–396, 2020.

[6] Y. Zhang, P. Li. and X. Wang, "Intrusion detection for IoT based on improved genetic algorithm and deep belief network," *IEEE Access*, vol. 7, no. 1, pp. 31711–31722, 2019.

[7] A. Yang, Y. Zhuansun, C. Liu, Li, J. and Zhang, C. "Design of intrusion detection system for Internet of Things based on improved BP neural network," *IEEE Access*, vol. 7, no. 1, pp. 106043–106052, 2019.

[8] R. Zarai, "Recurrent neural networks & deep neural networks based on intrusion detection system," *Open Access Library Journal*, vol. 7, no. 3, pp. 1, 2020.

[9] Q. Li, "A clustering and classify cation-based malicious attack detection method for Internet of Things," *Netinfo Security*, vol. 21, no. 8, pp. 82–90, 2021.

[10] Z. H. Shen, "Research on key technologies of attack detection based on machine learning for Internet of Things," Ph.D. dissertation, Jilin University, China, 2021.

[11] C. J. Han, "Design and implementation of Internet of Things DDoS attack traffic detection algorithm based on deep learning," M.S dissertation, Shandong University, China, 2020.

[12] H. R. Jin, "Research on abnormal traffic detection technology based on Internet of Things," M.S dissertation, Tianjin University of Technology, China, 2021.

[13] H. J. Chen, "Design and Implementation of collaborative anomaly detection method in Internet of Things environment," M.S dissertation, Tianjin University of Technology, China, 2021.