Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6856284, 9 pages
https://doi.org/10.1155/2021/6856284

Research Article

Intrusion Detection Analysis of Internet of Things considering Practical Byzantine Fault Tolerance (PBFT) Algorithm

Leixia Li,[1] Yong Chen,[2] and Baojun Lin[3]

[1] *University of Chinese Academy of Sciences, Beijing 100049, China*
[2] *Shijiazhuang Tiedao University, Shijiazhuang 050043, China*
[3] *Innovation Academy for Microsatellites of CAS, Shanghai 200000, China*

Correspondence should be addressed to Baojun Lin; linbj@microsate.com

Received 12 October 2021; Revised 15 November 2021; Accepted 22 November 2021; Published 11 December 2021

Academic Editor: Yuemin Ding

Copyright © 2021 Leixia Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to improve the security performance and accuracy of the Internet of things in the use process, it is necessary to use the Internet of things intrusion detection method. At present, the problem of inconsistency between the accuracy of detection results and nodes is more prominent when the Internet of things intrusion detection methods are running. This paper proposes a practical Byzantine fault-tolerant intrusion detection method for the use process of the Internet of things. This method introduces the intrusion detection method and the operation function of foreign attackers on the basis of practical Byzantine fault tolerance; using the expected utility function to the corresponding benefit function of practical Byzantine fault tolerance, the results of Internet of things intrusion detection model can be effectively calculated. Finally, the experimental results show that compared with the existing intrusion detection methods, the proposed method can effectively reduce the energy consumption of the Internet of things in the operation process, can effectively reduce 14.3% and 7.8%, and can effectively reduce the energy consumption of the Internet of things in the operation process.

1. Introduction

The Internet of things is a separate network established in the form of self-organization. The traditional Internet of things data can not meet people's monitoring needs compared with simple ushering in the era of Internet of things intrusion detection and analysis network [1–3]. Internet of things intrusion detection analysis is to add CMOS microcameras, microphones, and other facilities on the premise of using traditional Internet of things nodes to realize the performance of image acquisition, audio, video, and other data. Realize the application of the Internet of things intrusion detection and analysis network in the monitoring field, adopt the combination of the advantages of the traditional Internet of things and the Internet of things intrusion detection and analysis network, and use the coordination function of the combination to achieve the purpose of long-term, effective, and accurate monitoring in the environment. The system can realize the effect of real-time monitoring of a wireless multimedia network. When there are problems in the Internet of things or equipment on the network, the Internet of things intrusion detection system will receive early warning information, and the system will locate the fault information and inform the next level system [4–7].

In order to improve the efficiency of Internet of things intrusion detection and reduce the energy consumption in the use of Internet of things, this paper proposes an Internet of things intrusion detection method based on the practical Byzantine fault-tolerant algorithm. By giving priority to identifying nodes with high reliability for intrusion detection, all nodes in the network can be published at the end of detection, so as to realize the analysis of Internet of things intrusion detection.

2. Intrusion Detection Principle of Mobile Internet of Things

At present, the intrusion detection method for mobile network is to combine the principal component calculation method with fuzzy C uniform calculation to check the mobile network. The fuzzy c-means method is used to reduce the data aggregation in the mobile network, and the principal component method is used to reduce the data information after the cluster. The component aggregation of the data after dimension reduction needs to be compared with the corresponding dimension number and set the value. If the set value is the same as the comparison result, an alarm will be triggered to detect the mobile network.

Using the calculation method of principal component, the variable X of the sample is changed and substituted into the low-dimensional space Y [8, 9]. The formula is as follows:

$$Y = W^T X. \tag{1}$$

In the formula, the number of samples in the mobile network is represented by X, which is composed of the number of M observation objects and the number of N columns, and represents a value in the coordinates of M. The orthogonal photographic data combined by the sample covariance matrix is represented by W and calculated in the form of sample matrix. T stands for transpose matrix. C represents the covariance value of the sample.

$$C = N^{-1} \sum_{i=1}^{N} (X - u_i)^2, \tag{2}$$

where u_i is the average value and the following formula exists:

$$CW_i = \lambda_i W_i. \tag{3}$$

In the formula, $i = 1, 2, \cdots, N$ and W_i represent the i-th column sample covariance matrix existing in W in the orthogonal matrix, λ_i represents the special values occurring in matrix C, and W_i represents the special values represented by and special values. Arrange the data in order according to the size of the data, and combine the value λ_i corresponding to the first l special values to obtain the L -dimensional data after dimension reduction.

Set the random correlation matrix after u initialization to take value in the interval [0, 1], and meet the constraints of the following formula:

$$\begin{cases} \sum_{i=1}^{c} u_{ij} = 1, \quad \forall j = 1, \cdots, n, \\ \forall i, \forall j, u_{ij} \in [0, 1], \\ \forall i, \sum_{j=1}^{c} u_{ij} > 0. \end{cases} \tag{4}$$

Let v_i represent c cluster centers $i = 1, 2, \cdots, c$, and the calculation formula of cluster centerv_i is

$$v_i = \frac{\sum_{j=1}^{n} u_{ij} \cdot x_i}{\sum_{j=1}^{n} u_{ij}}, \quad \forall i. \tag{5}$$

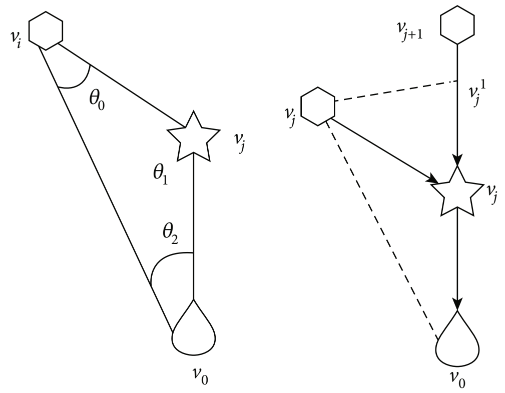


Figure 1: Distributed link design of IoT nodes.

Calculate the cluster center and the second from the formula. The Euclidean distance between this sample is d_{ij}.

$$d_{ij} = \left\| v_i - x_j \right\|. \tag{6}$$

The fuzzy optimal solution is obtained by the objective function J.

$$J = \sum_{i=1}^{C} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2. \tag{7}$$

In the formula, M represents the fuzzy weighting factor, and the objective function J represents the sum of squares of distances between each cluster center and each sample. Set the threshold ζ. When the target relationship value J is greater than the threshold ζ, prove that the sample data set is intrusion data, alarm, and complete the intrusion detection of mobile network.

3. Optimized Deployment of IoT Nodes and Information Fusion Processing

3.1. Optimized Deployment Design of IoT Nodes. In order to realize real-time feedback control of online learning and monitoring of things, firstly, the optimal structure design of single-networked nodes requires the use of balanced sensor node control methods for single-network node link distribution model structure, assuming that there are $k\,(k \geq 2)$ nodes in single network disjoint paths. Between$v$and$v\,(v)$, why the aggregate distribution of E, SN, RN nodes is$V = \{v_0, v_1, v_2,\cdots,v_n, \text{in}v_{n+1}, v_{n+m}\}$, the redundancy capacity of a single-layer relay node is described as $RC(v_j) = C(v_j) - w(v_i)$, and in Figure 1 in the defined multimedia monitoring area, the tree structure and cross structure are used to periodically forward data packets to the sensor node and cluster head node. The node dispersive link design that divides the time window between the sensor node and the cluster head node is presented.

As shown in Figure 1, the Internet of things is initialized through the monitoring node link model under multimedia, and the cluster head vector group D_n of the multimedia

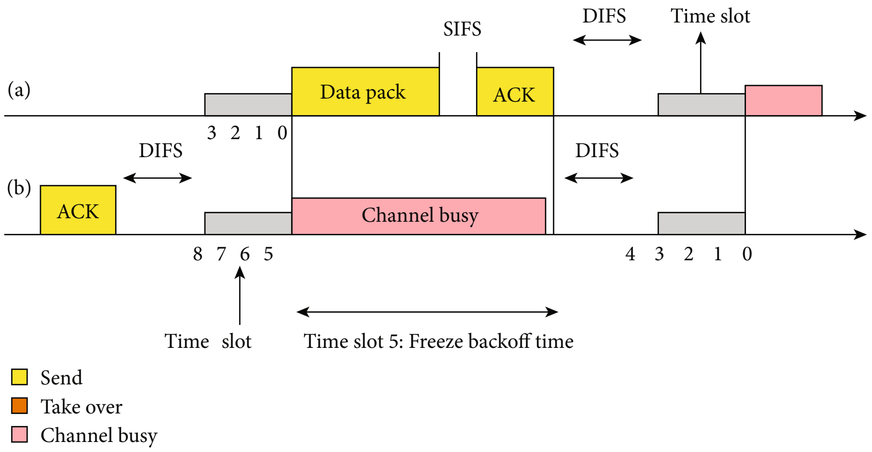


Figure 2: Layout of the channel allocation of two-layer relay nodes for multimedia monitoring.

sensing data node is constructed, and the data set is divided into multiple two-dimensional subregions A_k according to the width of the time window. A_k is mainly a two-dimensional information entropy group $|d_{n-\max} - d_{n-\min}| \cdot (1/K)$, which satisfies $A_k = A$. According to Figure 2, there is channel allocation layout of two relay nodes under the multimedia monitor.

If there is a certain space between the SN and sink of the Internet of things, $k = 1$ is initialized, and the position distribution set of multimedia sensor nodes under monitoring is $S = \{s_1, s_2, \cdots, s_n\}$, and the double-layer relay nodes are uniformly distributed. The layout method is constituted, using the obtained configuration feature distribution equation. The formula can be expressed by the following formula:

$$f(v_i, v_k) \geq 0, \tag{8}$$

$$\sum f(v_0, v_k) = 0 \quad (1 \leq k \leq n + m), \tag{9}$$

$$\sum f(v_i, v_k) = 0 \quad (1 \leq k \leq n), \tag{10}$$

$$\sum \left(f(v_j, v_k) - f(v_h, v_j)\right) = 0, \\ (n + 1 \leq k \leq n + m,\ 1 \leq h \leq n + m). \tag{11}$$

3.2. Information Fusion Processing of Sensor Nodes in the Cluster. Based on the detailed design of the corresponding configuration design of the Internet of things nodes, the data is extracted for the two-dimensional entropy feature corresponding to the sensor nodes in the cluster to alleviate the computational overhead information of monitoring multimedia [8]. The processing steps are as follows:

(Step 1) Enter the geographic coordinates of the Internet of things SN, sink, and initialize the operation of the scope of the Internet of things monitor.

(Step 2) Determine the sleep time. In the information data center, the distance $d = \{d_1, d_2, \cdots, d_n\}$ between SN and sink is sorted. The group corresponding to the cluster location distribution of multimedia sensor nodes in the monitoring area is $S = \{s_1, s_2, \cdots, s_n\}$.

(Step 3) $k = 1$ is initialized, the pseudorandom number adaptive sorting of the multimedia sensor sequence is determined from the current position, and Tag = 1 is placed.

(Step 4) When $d_k \geq d_0$ and Tag$(k) = 1$, obtain the best position of the data cluster fusion center and move to step 5. Otherwise, the algorithm ends.

(Step 5) Setting the maximum number of hops for the learning factor s_k of the Internet of things. count_max $(s, v)\chi_k$ is associated with the sensing IoT route itself and Tag$(k) = 0$.

$$\left| \frac{d(s_k, v_0)}{d_0} - \chi_k \right| \leq 0.5. \tag{12}$$

(Step 6) Select nodes randomly arranged on the connection line of s_k and sink under the sensing Internet of things, and adjust the energy and resources of a network node calculated by RN in the monitoring area $(\chi_k - 1)$ under the sensing Internet of things, and set s_k. The next hop is marked as k, and the multimedia detection information output by the sink node in the range of $d0$ is quantified, fused, tracked, and identified.

3.3. Optimization of IoT Learning Monitoring Feedback

3.3.1. Channel Balanced Allocation Design of the Internet of Things. In any state error under the sensor IoT node, the node variable can be defined as$f(T)$ and, at the same time, clarify the different characteristics of random nodes in the cluster.

$$\chi_k \le \text{hop\_count\_max}\ (s_k, v_0)\left|\frac{d(s_k, v_0)}{d_0} - \chi_k\right| \le 0.5. \tag{13}$$

The threshold for selecting data is $\{\text{xmax}, \text{xmin}\}$. By defining the connection line between each node SN and sink in the cluster, the maximum number of hops can be obtained and the node dimension entropy of IoT learning can be obtained. Meet the following formula:

$$d(v_i, v_0) = d\left(v_i', v_0\right). \tag{14}$$

The average value of data near the monitoring node learned through the Internet of things has the spatial characteristics of the distribution of monitoring feedback data

$$\begin{aligned} d\left(v_i', v_j'\right) &> \frac{1}{2} d\left(v_{i+1}, v_j\right); d\left(v_i', v_j\right) \\ &= \frac{1}{2} d\left(v_{i+1}, v_j\right); d\left(v_i', v_j\right) < \frac{1}{2} d\left(v_{i+1}, v_j\right). \end{aligned} \tag{15}$$

The information probabilities l and g of the unit data subset are integers, assuming that the probability weighted distance of each node determines the threshold group. After $d(v_i', v_j) > (1/2)d(v_{i+1}, v_j)$, the channel allocation control function used for data transmission in a single node i has been described.

$$\begin{gathered} d\left(v_{i+1}, v_j\right)/d_0 = l\frac{d\left(v_{i+1}, v_j\right)}{d_0} = l, \\ \frac{d\left(v_i, v_j\right)}{d_0} \longrightarrow \lambda; \left[\frac{d\left(v_i, v_j\right)}{d_0}\right] + 1 = \lambda + 1, \\ \frac{d\left(v_{i+1}, v_j\right)}{d_0} = \lambda \cdot \gamma. \end{gathered} \tag{16}$$

3.3.2. IOT Transmission Delay Control. The linear shift channel allocation method is used, and the intelligent search algorithm for the transmission delay control of the Internet of things is used to perform the processing of the Internet of things learning monitoring feedback link equalization.

$$e = \frac{1}{n\_\text{ever}} \sum_{i=1}^{n\_\text{ever}} \text{sqrt}\left((x_i - x\wedge_i)^2 + (y_i - y\wedge_i)^2\right), \tag{17}$$

$$\text{hop\_count\_max}\ (v_i, v_o) = \left[\frac{d(v_i, v_0)}{d_0} + 1\right]. \tag{18}$$

When the distance from the SN to the sink is arranged in descending order and the load of the object's network cluster head n_i is constant, the global balanced scheduling method is used to perform real-time feedback control of monitoring information. The fuzzy adaptive weighted control processing of multimedia sensor nodes describes the critical thresholds related to physical network learning to monitor real-time feedback.

$$E_{Tx}(L, d) = \begin{cases} LE_{\text{elect}} + L\varepsilon_{fs}d^2, & d < d_0, \\ LE_{\text{elect}} + L\varepsilon_{mp}d^4, & d > d_0, \end{cases} \tag{19}$$

$$E_{Rx}(L) = LE_{\text{elect}}. \tag{20}$$

In the formula, E_{elect} represents the global energy equalization coefficient. Through the above processing, the global equalization control of the transmission link of the Internet of things is realized, and the real-time feedback capability of multimedia information learning and monitoring is improved through the optimal configuration of the smart phone node.

3.3.3. IoT Performance Monitoring Model. RTFM is a working group established by IETF. We have proposed a general framework for describing and measuring single network services. And based on this framework, based on RTFM, the Internet of things intrusion detection and real-time risk based on the real-time monitoring platform of the Internet of Thousands of Things are proposed. The early warning and real-time risk early warning system model is shown in Figure 3. The model is classified into four modules: rule input system, flow collection system, data analysis system, and database system.

In this model, the traffic collection system is based on the rules set by the rule input system. Thousands of real-time Internet of things traffic filter and aggregate the Internet of things traffic, save the effective data in the database system, and provide it for data analysis. The system handles it. The data analysis system processes valid data to obtain the changes and distribution information of the Internet of things intrusion to predict, adjust, and manage the actions of the Internet of things.

The flow of the system model is shown in Figure 4.

In the shared media Internet of things, any packet that flows through the Internet of things is higher than the grouping required by the hardware configuration of the Internet of things segment business, making it impossible to process the intercepted packets in time. As long as the server used for IoT intrusion analysis is installed in the network segment interconnected with the outside world and the network card of the machine is set to "hybrid" mode, all IP data packets entering and leaving the Internet of things can be captured, and if the IP packets are analyzed and then compiled if successful, you can get the necessary information such as the source address, destination address, data volume, and application protocol. Its advantage is that it does not change the structure of the Internet of things, does not increase the load of the Internet of things, and does not occupy the resources of the Internet of things. It has nothing to do with the waiting time of the Internet of things and does not affect the network usage of user items.

This article uses Raw Socket to implement the Sniffer method is relatively simple but only cuts the packet above the IP layer and does not contain frame information. It can not meet some special requirements. From the analysis of the current Internet of things intrusion model, it can be seen that the entire Internet of things intrusion is mainly the practical Byzantine Fault Tolerance (PBFT) algorithm traffic, and the changes in the practical Byzantine Fault Tolerance (PBFT) algorithm traffic basically reflect the changes in the entire Internet of things intrusion, so it can use the Practical Byzantine Fault Tolerance (PBFT) algorithm traffic instead of total traffic to analyze the performance of the Internet of things; that is, you can use Raw Socket to obtain traffic information.

3.4. Practical Byzantine Fault Tolerant Intrusion Detection Method

3.4.1. Practical Byzantine Fault Tolerance Algorithm. A practical Byzantine fault-tolerant algorithm is constructed through four tuples, the income function of attacker and intrusion detection system is obtained according to the model, the attack strategy group and defense strategy group are constructed according to the income function, the income matrix of the game model is obtained by using the desired function, and the Nash balance of the game model is calculated according to the income matrix.

The representative is the practical Byzantine fault-tolerant algorithm, which represents the model R_{RDM} through the four tuple attender, action, profits, and times.

$$R_{\mathrm{RDM}} = (\text{attender, action, profits, times}). \tag{21}$$

In the formula, attender has intrusion detection system and intruder. The attacker is replaced by a and the intrusion detection system is represented by B. The intrusion detection system is different from the attacker's attack space: attacker a's attack space. $A_a = (N, A, M, P)$ includes normal, attack, abnormal, and preattack. Formula $A_d = (C, R, W, D)$ represents the action space of intrusion detection system. It includes continuous execution, execution, early warning, and protection.

Set the respective representatives of $U_a(A_a)$ and $U_d(A_d)$ as the revenue function of attacker and intrusion detection system, T represents the total number of games, and equation (21) was converted into the following equation:

$$R_{\mathrm{RDM}} = (a, d\,; A_a, A_d\,; U_a(A_a), U_d(A_d)\,; T). \tag{22}$$

Make attack strategy suit and defense strategy suit according to formula (22). The expressions are as follows.

$$S_a = (S_N, S_M, S_P, S_A), \tag{23}$$

$$S_d = (S_C, S_R, S_W, S_D). \tag{24}$$

In the formula, S_N, S_N, S_P, and S_A represent normal, attack, abnormal, and preattack action strategies, respectively. $S_C, S_R, S_W,$ and S_D, respectively, indicate continued execution, recommended execution, alarm, and protective action. $S_{ad} = (s_a, s_d | s_a \in S_a, s_d \in S_d)$ expressed the action strategy of both sides of the bureau.

In the case of action strategy $S_{ad} = (S_A, S_C)$, the attacker obtains the highest benefit in the mobile network. In action strategy $S_{ad} = (S_P, S_D)$, the intrusion detection system is beneficial to the attacker when the attacker attacks the moving body [10, 11]. If an attacker wants to attack a moving body on the network, the intrusion detection system will take preventive measures. Through the above analysis, the intrusion detection system and the attacker's action strategy are converted into the corresponding preference set, and the expected utility function is used to set the preference set in the interval $[0, 1]$. $U(X)$ represents the effective function of both sides of the game. The behavior of $U\,(X)$ is as follows:

$$U(X) = \sum_{S_{ad}} P_1 u(x_1) + \cdots + P_k u(x_k). \tag{25}$$

The equation represents the probability of intrusion detection system or attacker adopting different strategies in mobile network each time and represents the benefits of various strategies. Under ideal conditions, the probability of different action processing by attacker and intrusion detection system is 0.25. $u(x_i)$ can calculate the income matrix of the game model and get the balance of the game model.

3.5. Optimization Method of Practical Byzantine Fault-Tolerant Intrusion Detection Method. The adjustment value of the balance calculation method can be fed back randomly, the practical performance of the mobile network is increased by the adjusted probability of P_i, the profit factor P_i is substituted into the function formula, and the intrusion detection method is actually optimized. The setting of $P^* = (P_1^*, P_2^*, \cdots, P_i^*)$ represents the random optimal equilibrium value that occurs in the real Byzantine calculation method. $P_i^* = (P_1^*, P_2^*, \cdots, P_k^*)$ represents the situation of intrusion detection system and tactics used by attackers in a game. P_i^* represents the tactical hybrid probability set by the intrusion detection system and the attacker. Δ_i represents a collection of hybrid tactics.

$$\Delta_i = \{P_i^* = (P_1^*, P_2^*, \cdots, P_k^*)\}. \tag{26}$$

The mixed strategic space $\Delta = \prod \Delta_i$ of the intrusion detection system and the attacker is obtained by equation (26). Through the strategy space Δ, the probability function π_i^* of the intrusion detection system and the attacker's choice of action strategy is obtained.

$$\pi_i^* = \frac{\lambda u(x_i)}{\sum_{k=1}^{i} \lambda u(x_i)}. \tag{27}$$

In the formula, λ denotes the weighting coefficient. When λ approaches infinity, the stochastic optimization reflects that the equilibrium is close to nanometer equilibrium, and the final result can be obtained by equation (26).

The ideal probability in a normal mobile network is almost zero, and attackers use different attack methods and strategies to attack the mobile network. Formulas (1) and (14) are optimized consistently to improve the detection accuracy of the actual Byzantine intrusion detection method. Suppose $\delta \in [0, 1]$ is representative of the profit factor. The profit of both parties during the game is determined by the profit factor. The larger the value of δ, the more important the intrusion detection system or the attacker attaches to the overall profit and the higher the rate of return. The smaller value ofδ indicates that the intrusion detection system or the attacker pays less attention to the overall income. If the efficiency function is imported, the following formula can be obtained:

$$U(X) = \sum_{k=1}^{\infty} \left(1 - \pi_i^*\right)^{k-1} \pi_i^* \delta^{k-1}. \tag{28}$$

Formula (28) belongs to the revenue function. Players in each game need to actively change their action tactics to maximize the revenue value, check malicious attacks in the mobile network according to the revenue function, and delete the combination of malicious nodes to improve the security performance of the mobile network [12–14].

3.6. Disciplinary Mechanisms. The use of disciplinary agencies poses a threat to malicious nodes in the mobile network. The purpose of the retribution mechanism is to have one node representing the attack, but the other nodes are not represented by the malicious node transmitted from the next timeslot. There are three levels of punishment.

(1) If no malicious node is detected in the previous mobile network game, all nodes remain in the current state in the network, and if the malicious node is detected in the mobile network, it will move to the next step

(2) Punishment agencies punish malicious nodes and keep other nodes in the mobile network in their original state during punishment

(3) After malicious node operation, the real data $(U_1, \cdots, U_N)$ of mobile network is shown below. If there is malicious behavior in the third step, it needs to go back to the second step according to the punishment malicious node

If the malicious node ϑ becomes normal, the maximum profit of node ϑ in the departure time slot is $\bar{v}_k$, the profit of node ϑ in the punishment T_k time is $\dot{v}_k$, and the profit of node ϑ in the normal state is v_k'. The average discounted utility value $\widehat{U}_k$ of node ϑ in mobile Internet of things can be obtained:

$$U_k = (1-\delta)\bar{v}_k + \delta\left(1 - \delta^{T_k}\right)\dot{v}_k + \delta^{T_{k+1}} v_k'. \tag{29}$$

The utility value of the average discount of the node ϑ in the game when the node ϑ is in the normal state is

$$U_k = (1-\delta) \sum_{t=0}^{\infty} \delta v_k = v_k \tag{30}$$

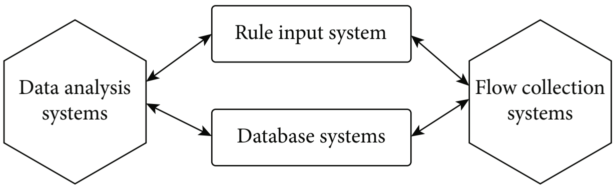


Figure 3: System model.

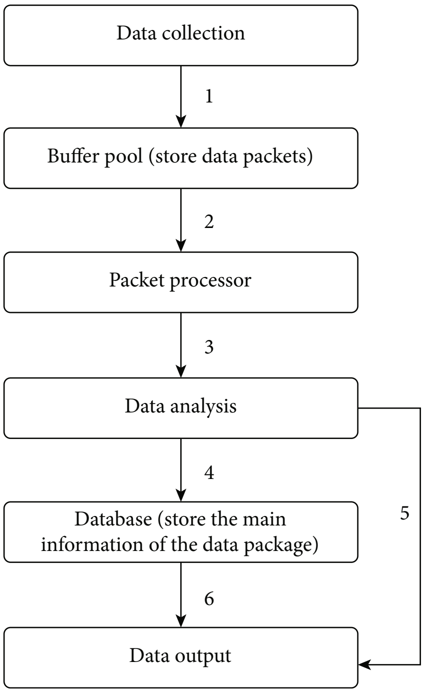

Data collection
1
Buffer pool (store data packets)
2
Packet processor
3
Data analysis
4
Database (store the main information of the data package)
5
6
Data output

Figure 4: System process.

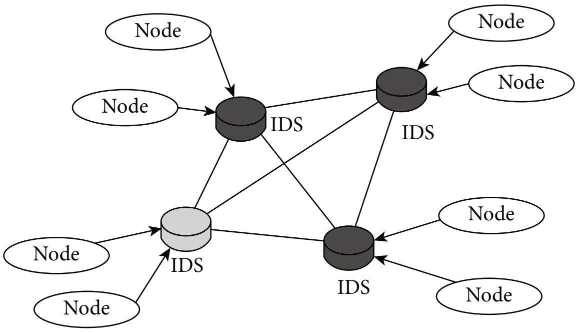


Figure 5: Network topology.

According to Formulas (29) and (30), the deviation profit ΔU_k of node ϑ in mobile network can be obtained. ΔU_k is calculated by the following formula:

$$\Delta U_k = \widehat{U}_k - U_k. \tag{31}$$

In formula (31), the deviation gain must be below zero, and the deviation gain of node ϑ is smaller than the cooperation gain. At this point, no rational node in the mobile network deviates from the normal state.

Table 1: Attack types.

Attack types	Attack subtype
Normal	Normal
DoS	Back, land, Neptune, pod, smurf, teardrop, apache2, adpstorm, processtable, worm
Probe	Ipsweep, nmap, portsweep, satan, mascan, saint
U2R	buffer_overflow, loadmodule, perl, rootkit, perl, sqlattack, xterm, ps

4. Experimental Results and Analysis

4.1. Network Topology. The network cluster structure adopted in this paper is mainly composed of cluster head node and general node, which is a general topology structure in practical application. The common nodes, namely, sensor nodes, are terminal nodes in single network. The cluster head node (that is, the gateway device in a single network) manages the nodes in the cluster and reports data to the outside world. IDS operates at the cluster head node, and each cluster head node performs intrusion detection between it and other cluster head nodes based on PBFT [15]. The specific network topology is shown in Figure 5.

4.2. Data Set Preprocessing. The KDD Train of the NSL-JDD data set was tested experimentally. The training model of 20 percent training set allows nodes to randomly select one from the test subset of KDD test-21 to record and perform intrusion detection operations as the node at this time (Table 1).

4.3. Experimental Evaluation Criteria. The security performance of intrusion detection method is mainly reflected in the detection rate indicator, especially the ratio of the number of detected malicious nodes to the total number of malicious nodes in the network. Common evaluation criteria are used here: (1) TP (True Positive) indicates the number of samples that are correctly judged as positive types, (2) TN (True Negative) indicates the number of samples that are correctly identified as Negative, and (3) FP (False Positive) indicates the number of samples whose sample error is judged to be negative.

Then, Detection Rate (DR) is defined as

$$\mathrm{DR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}}. \tag{32}$$

The other mode of intrusion detection is energy consumption, which has the characteristics of the following: (1) EV is the energy consumption in node election, (2) EDI is the energy consumption of node intrusion detection, (3) EP is the energy consumption of node to publish detection results, and (4) EC is the statistical energy consumption of node measurement results.

Energy consumption of all nodes in the network is

$$E = \sum_{i}^{N} \mathrm{EV}_i + \sum_{i}^{N'} (\mathrm{EID} + \mathrm{EP}) + \sum_{i}^{N} \sum_{j}^{N''} \mathrm{EC}_{ij}. \tag{33}$$

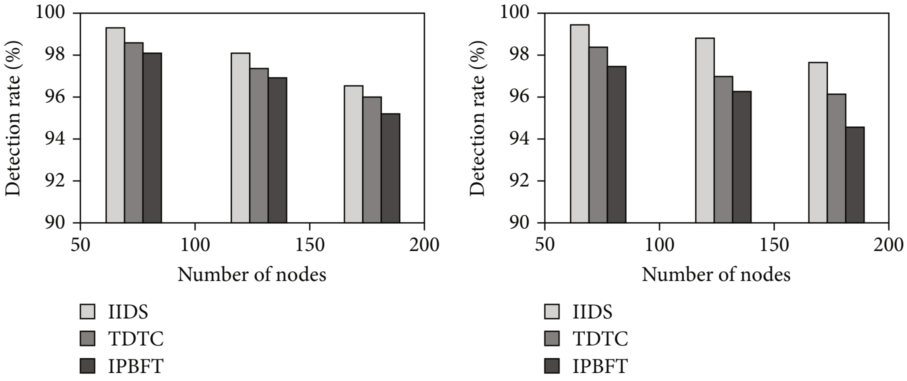
Table 2: Simulation parameters.

Parameter	The default value
Network area size	400×400
Number of nodes	60~300
Number of abnormal nodes	15%-20%
Initialize weights w_i	1
Initialize the trusted list	node_i, $i \in (0, N-1)$
Detection interval Δt	70 s
The elapsed time	20 min

4.4. Experimental Scheme Design. In order to test the effectiveness of the practical Byzantine fault tolerance algorithm proposed by ontology, the SVM algorithm is used to detect the NSL-KDD data set to obtain the detection rules. In the experimental process of this paper, the relevant rules of flow control and intrusion detection can be realized by using the microcontroller, and the Active Message layer can be used to effectively control the RF module, so as to realize the mutual communication between the communication nodes. Based on the previous single network intrusion detection methods, this paper presents a comparative experimental study. Detailed simulation environment parameters are shown in Table 2.

4.5. Analysis of Results. In this paper, simulation experiments are conducted on different types of network nodes and abnormal proportion nodes in the initial state, as shown in Figure 6. Experimental results obtained in different types of network states can be seen. As can be seen from the experimental results in Figure 6, due to the impact of dimensionality reduction, the accuracy of the training data set will also be affected to a certain extent. Therefore, the two-dimensional reduction detection method is used to optimize the model used by the algorithm in this paper to ensure that the detection rate of colleges and universities can be achieved under the condition of a large number of network nodes.

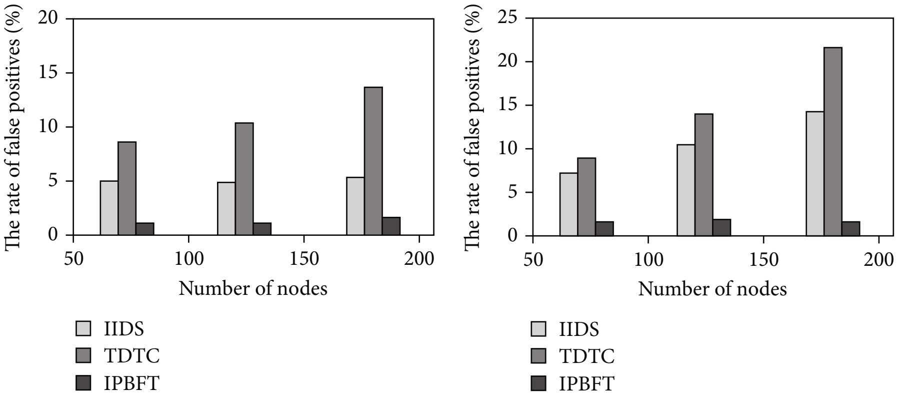
Figure 7 shows the error rate in a single network with different proportions of the three methods attacking nodes. As can be seen from Figure 7, PBFT can modify the detection errors of a single node, and the introduced matching protocol has the lowest error rate. However, false positives occur on the network only when detection errors occur on nodes above $M + 1$. On the other hand, IIDS and TDTC methods only rely on the detection results of a single node, especially the TDTC method which has the highest false



(a) There are 10% attack nodes in the network

(b) There are 15% attack nodes in the network

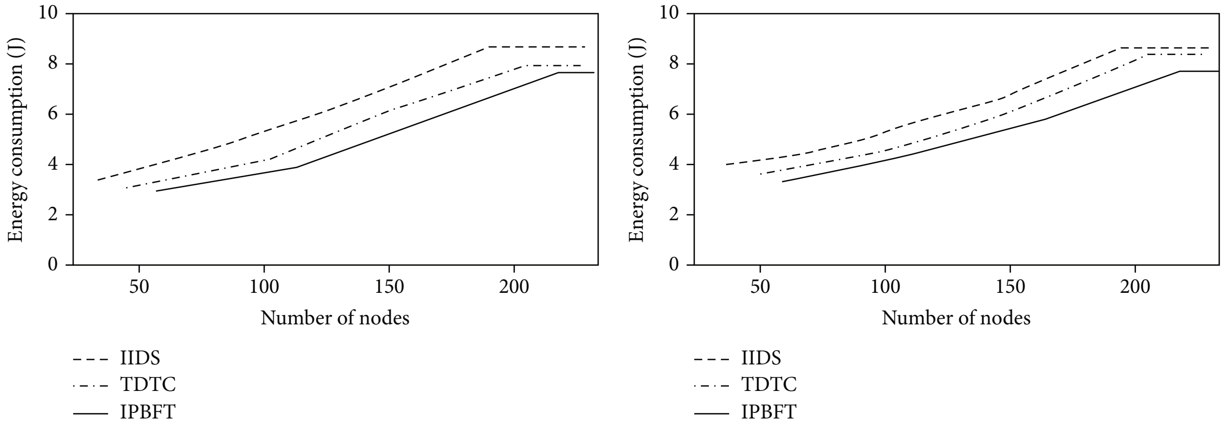
Figure 6: Detection rates of three intrusion detection methods under different proportions of abnormal nodes.



(a) There are 10% attack nodes in the network

(b) There are 15% attack nodes in the network

Figure 7: False positive rate of three intrusion detection methods under different proportions of abnormal nodes.



(a) There are 10% attack nodes in the network

(b) There are 15% attack nodes in the network

Figure 8: Energy consumption of three intrusion detection methods for different proportions of abnormal nodes.

positive rate of the three methods after the accuracy of the data set is reduced.

By evaluating the network intrusion detection method, the network energy consumption can be obtained. Under the same experimental conditions, the algorithm proposed in this paper can complete the algorithm test under the environment of as little energy consumption as possible. As shown in Figure 8, after setting 15% and 20% of the total number of different nodes for different attackers, the energy consumption of all nodes in the experimental system is compared and analyzed to ensure that the energy consumption of IPBFT on network nodes is as low as possible. Compared with the IDS algorithm, the energy consumed by IPBFT algorithm can be reduced to 13.5% when the attacking node process takes up 25%, and the network attacking node can also be reduced from 20% to 12.5%. Compared with the TDTC algorithm, IPBFT algorithm can reduce energy consumption by 6.9% when attack nodes account for 15% of network usage. Compared with the other two methods, the IPBFT algorithm in this paper consumes less energy.

5. Conclusions

The Internet of things has been widely used in many fields such as life service, machinery manufacturing, medical treatment, economy, and business, but it will lead to the loss of data and information and lead to serious losses when it is invaded by external hackers, viruses, and viruses. In existing mobile Internet of things in the intrusion detection, test results are not accurate, and node inconsistent problem, through the practical Byzantine fault tolerance in the mobile Internet of intrusion detection method, can effectively solve the serious problems that exist in the existing methods, through the experimental results which show that the method can effectively improve the safety and accuracy of the mobile Internet of things.

Data Availability

Data sharing is not applicable to this article as no data sets were generated or analyzed during the current study.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This study is sponsored by Innovation Academy for Microsatellites of CAS.

References

[1] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, “Internet of things intrusion detection: centralized, on-device, or federated learning?,” *IEEE Network*, vol. 70, no. 5, pp. 5057–5070, 2020.

[2] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, “A survey of intrusion detection in Internet of things,” *Journal of Network & Computer Applications*, vol. 68, no. 4, pp. 4089–4093, 2017.

[3] M. Ge, N. Syed, X. Fu, Z. Baig, and A. Robles-Kelly, “Toward a deep learning-driven intrusion detection approach for Internet of things,” *Computer Networks*, vol. 21, no. 7, pp. 30–35, 2020.

[4] P. Nimbalkar and D. Kshirsagar, “Feature selection for intrusion detection system in Internet-of-things (iot),” *ICT Express*, vol. 7, no. 2, pp. 177–181, 2021.

[5] J. Balasundaram and M. Pushpalatha, “A novel optimized bat extreme learning intrusion detection system for smart Internet of things networks,” *International Journal of Communication Systems*, vol. 6, no. 4, pp. 6125–6133, 2021.

[6] R. Fantacci, F. Nizzi, T. Pecorella, L. Pierucci, and M. Roveri, “False data detection for fog and Internet of things networks,” *Sensors*, vol. 19, no. 19, pp. 4235–4507, 2019.

[7] F. Medjek, D. Tandjaoui, I. Romdhani, and N. Djedjig, “Message from the SmartData-2017 steering chairs,” in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 735–742, Exeter, United Kingdom, 2017.

[8] G. Thamilarasu, A. Odesile, and A. Hoang, “An intrusion detection system for Internet of medical things,” *IEEE Access*, vol. 8, pp. 181560–181576, 2020.

[9] A. J. Siddiqui and A. Boukerche, “Tempocode-iot: temporal codebook-based encoding of flow features for intrusion detection in Internet of things,” *Cluster Computing*, vol. 24, no. 1, pp. 17–35, 2021.

[10] M. Ahmad, Q. Riaz, M. Zeeshan, H. Tahir, and M. S. Khan, “Intrusion detection in Internet of things using supervised machine learning based on application and transport layer features using unsw-nb 15 data-set,” *EURASIP Journal on Wireless Communications and Networking*, vol. 73, 627 pages, 2021.

[11] A. Bamou, “Intrusion detection in the Internet of things,” *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 1.5, pp. 1–7, 2020.

[12] A. Yang, H. Liu, Y. Chen, C. Zhang, and K. Yang, “Digital video intrusion intelligent detection method based on narrowband Internet of things and its application,” *Image and Vision Computing*, vol. 97, no. 4, pp. 103914–103957, 2020.

[13] J. Arshad, M. A. Azad, M. M. Abdellatif, M. H. U. Rehman, and K. Salah, “Colide: a collaborative intrusion detection framework for Internet of things,” *IET Networks*, vol. 8, no. 1, pp. 3–14, 2019.

[14] S. Halder, A. Ghosal, and M. Conti, “Efficient physical intrusion detection in Internet of things: a node deployment approach,” *Computer Networks*, vol. 154, no. MAY 8, pp. 28–46, 2019.

[15] S. Deshmukh-Bhosale and S. S. Sonavane, “A real-time intrusion detection system for wormhole attack in the rpl based Internet of things,” *Procedia Manufacturing*, vol. 32, pp. 840–847, 2019.