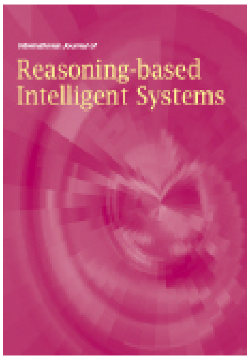


International Journal of Reasoning-based Intelligent Systems

ISSN online: 1755-0564 - ISSN print: 1755-0556
https://www.inderscience.com/ijris

A privacy protection method for IoT nodes based on convolutional neural network

Yuexia Han, Di Sun, Yanjing Li

DOI: 10.1504/IJRIS.2023.10053461

Article History:

Received:	22 September 2022
Accepted:	08 November 2022
Published online:	19 March 2024

A privacy protection method for IoT nodes based on convolutional neural network

Yuexia Han, Di Sun* and Yanjing Li

Department of Information Engineering,
Shijiazhuang University of Applied Technology,
Shijiazhuang 050081, China
Email: hyx1209@126.com
Email: sundi721721@sina.com
Email: 532465444@qq.com
*Corresponding author

Abstract: In order to improve the security of internet of things, a privacy protection method of internet of things nodes based on convolutional neural network is proposed. Firstly, the flow model of IoT network nodes is constructed while using the ant colony algorithm to solve the model to obtain the current flow data of IoT nodes. Secondly, a convolutional neural network model is established to identify abnormal nodes in the internet of things. Finally, the privacy protection strategy of k-anonymous IoT nodes based on the average degree of nodes is adopted to protect the privacy of IoT abnormal nodes. The experimental results show that the method can effectively extract the node traffic before and after the attack on the internet of things, and the deviation value is only 2 kb/s; the identification results are more accurate, and the privacy of the internet of things nodes can be effectively protected.

Keywords: convolutional neural network; internet of things; IoT; node privacy; protection method; anonymity.

Reference to this paper should be made as follows: Han, Y., Sun, D. and Li, Y. (2024) ‘A privacy protection method for IoT nodes based on convolutional neural network’, *Int. J. Reasoning-based Intelligent Systems*, Vol. 16, No. 1, pp.16–25.

Biographical notes: Yuexia Han received her Master’s degree in Computer Applied Technology from North China Electric Power University in 2007. She is currently an Associate Professor in the Department of Information Engineering of Shijiazhuang University of Applied Technology. Her research interests include internet of things application, information security and artificial intelligence.

Di Sun received her Master’s degree in Computer Technology from Hebei University of Science and Technology in 2018. She is currently a Lecturer in the Department of Information Engineering of Shijiazhuang University of Applied Technology. Her research interests include artificial intelligence, computer network and image processing.

Yanjing Li received her Master’s degree in Computer Applied Technology from Shijiazhuang Tiedao University in 2012. She is currently a Lecturer in the Department of Information Engineering of Shijiazhuang University of Applied Technology. Her research interests include Internet of things, network and information security.

1 Introduction

With the development of network technology, network software and websites emerge in endlessly, such as QQ, microblog, Facebook, etc., making the number of network users grow exponentially. As one of the technical frameworks of the current network interaction environment (Bica et al., 2019), the internet of things (IoT) has a wide range of applications. The internet is used in various industries to provide network services for different users. The continuous development of the IoT technology has accelerated the pace of the IoT era, and has also made the number of terminals connected to the IoT increase rapidly, which has transformed the current network environment data from a single production data to a data environment where production and use data coexist. Among the large amount of data generated by users’ behaviours on social networks, they have strong business value and play a great role in their application scenarios. In the process of network interaction, users exchange data through IoT nodes. IoT nodes are dynamic parts of network communication transmission and are weak. When they are attacked maliciously by network hackers (Park, 2020), it is easy to cause users’ privacy data leakage. By analysing the characteristics of users’ privacy data (Meng et al., 2019), network hackers can tap into the commercial, social and other values behind users’ privacy data, Its application in a bad environment will cause unpredictable consequences.

Therefore, in order to improve the privacy security of IoT nodes, it is essential to protect the privacy of IoT nodes, which is of high practical significance.

Zhang et al. (2021) proposed the security and privacy protection methods of IoT collaborative edge computing. This method is based on the architecture of the IoT, uses the write edge computing method to calculate the security of the current IoT communication nodes, and then uses the DAT data protection method to achieve the protection of data nodes in the communication transmission process of the IoT. However, in the application process of this method, it needs to use sensors to collect the node queues in the current IoT communication. There is delay and interference noise when sensors collect nodes, which makes it invalid to protect the privacy of IoT nodes. Li et al. (2019b) proposed a network data protection algorithm. This method realises the privacy protection of IoT nodes by building the dynamic topology of the IoT, setting the time window, using hierarchical sampling to obtain the current IoT communication nodes, and using sensor network data protection methods to hide the IoT communication network topology. However, this approach is different from the IoT framework. When building the dynamic topology of the IoT, users need to input the attributes of the IoT framework themselves. The process of building the dynamic topology of the IoT is tedious, so its application scope is very small. Dimitriou and Roussaki (2019) proposed location privacy protection methods in the IoT environment. This method protects the privacy of location information when users communicate. This method obtains the current IoT communication nodes by using the clustering method, and then establishes the attacker model, uses this model to obtain the attacking nodes in the cluster IoT communication, and then uses the data propagation protocol to achieve the privacy protection of user location communication nodes. Although the above methods can complete the privacy protection of IoT nodes, there are problems such as insufficient accuracy of node traffic identification before and after the IoT is attacked, and low accuracy of IoT node loss value.

In order to effectively improve the privacy security of IoT nodes, a privacy protection method of IoT nodes based on convolutional neural network is proposed. The overall technical route of this method is as follows:

1. According to the hierarchical structure of the IoT and its deep coupling relationship with the user communication network, the node flow model of the IoT is established.
2. After solving the node traffic model of the IoT using ant colony optimisation algorithm, the current node traffic data of the IoT is obtained.
3. Input the current IoT network node traffic data into the convolutional neural network model. The convolutional neural network model identifies the abnormal node in the current IoT network node traffic by establishing the IoT network node traffic cube sample matrix, feature extraction, sliding convolution and other operations. This node is also the node that is under malicious attack.
4. The malicious IoT node is taken as the privacy protection object, and a k-anonymous IoT node privacy protection policy based on node average is proposed.

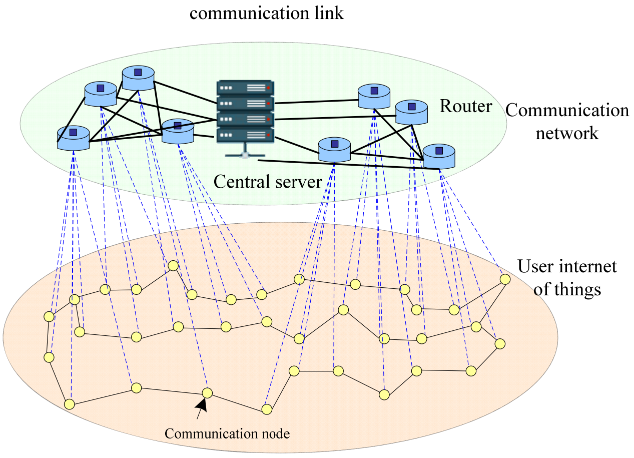
2 Privacy protection method of IoT nodes based on convolutional neural network

2.1 Construction and solution of node traffic model of IoT network

The structure of the IoT is a layered form, and there is a deep coupling relationship between it and the user's communication network. The relationship between the IoT and the user communication network is shown in Figure 1.

If there is a malicious attack during the interaction between the IoT and the user communication network, it will lead to changes in network traffic between the IoT and the communication network (Liu et al., 2021; Li and Wu, 2020; Taguchi and Yoshimura, 2021).

Figure 1 Schematic diagram of the relationship between IoT and user communication network (see online version for colours)



According to the relationship between the IoT and the user communication network in Figure 1, the extreme learning machine algorithm is used as a fitting function to establish a network node traffic model of the IoT, and the model is used to obtain the network node traffic when the IoT communicates with users. The detailed process is as follows:

Let G = {(x_1, t_1), (x_2, t_2), …, (x_N, t_N)} represent the traffic dataset between the IoT and the communication network, where $x_i = [x_{i1}, \ldots, x_{in}] \in R^n$, $t_i = [t_{i1}, \ldots, t_{in}] \in R^m$, and i = 1, 2, …, N, the expression formula of the extreme learning machine regression model of the network traffic dataset is as follows:

$$\sum_{i=1}^{L} O_i g * \sum_{i=1}^{L} \left(I_i * x_j + b_1 + L_1 \right) = t_j = t_j \tag{1}$$

In the above formula, L and b represent the number and threshold of hidden layer nodes respectively; I_i, β_i represent the node weights of input layer and hidden layer respectively; g(x) is the excitation function.

Convert formula (1) into matrix form, and its expression formula is as follows:

$$HO = T \tag{2}$$

In the above formula, H represents the input matrix of the hidden layer, O represents the partial derivative matrix, $O = \begin{bmatrix} O_1^T \\ O_2^T \\ \vdots \\ O_L^T \end{bmatrix}_{L*m}$; t represents the network traffic time matrix, $T = \begin{bmatrix} T_1^T \\ T_2^T \\ \vdots \\ T_L^T \end{bmatrix}_{N*m}$.

The nonlinear regression matrix problem of formula (2) is converted into equation form, and its expression formula is as follows:

$$L = \operatorname{argmin}\left(\frac{1}{2}\|O\|^2 + \frac{\gamma}{2}\|\varepsilon\|^2\right)\sum_{i=1}^{L} O_i g\left(I_i x_j + b_i\right) - t_j - \varepsilon_j \tag{3}$$

In the above formula, $\|\varepsilon\|^2$ represents extreme learning and structural risk; γ is the adjustment parameter.

Reduce formula (3), and then use Lagrangian function to solve the result. The expression formula is as follows:

$$L\left(O', \varepsilon, \omega\right) = \frac{\|O\|^2}{2} + \frac{\gamma * \|\varepsilon\|^2}{2} - \omega H O^{-T} - \omega\varepsilon \tag{4}$$

In the above formula, ω represents the Lagrange multiplier.

Calculate the partial derivative of formula (4), and its expression formula is as follows:

$$O' = \left(H^T H * H^T T + \frac{H^T T}{\gamma}\right)^{-1} \tag{5}$$

After the above steps, the expression formula of the network node traffic model between the IoT and the communication network based on the extreme learning machine is as follows:

$$\hat{t} = \sum_{o=1}^{L} O_i g \sum_{o=1}^{L} \left(I * x + b_i\right) \tag{6}$$

So far, the network node traffic model of the IoT has been built.

Ant colony algorithm is a kind of global search algorithm (Gao et al., 2019), which is inspired by the process of ant colony searching for food in nature. From a mathematical point of view, the ant colony algorithm has complexity and randomness. It is a process of iteration step by step until the algorithm reaches the final convergence by initialising the number of ants and updating the number of pheromones (Lakshmanaprabu et al., 2019). The ant colony algorithm has a high execution efficiency. When establishing the ant colony optimisation path, it can build its optimisation path according to the probability of the next access object. The algorithm has good convergence, and its output results are more accurate (Li et al., 2019a). In the final summary, the ant colony algorithm is used to solve the node traffic model of the IoT. The specific process is as follows:

Let m represent the total number of ants in the ant colony, i and j both represent the flow nodes, and the distance between them is represented by d_{ij} (i, j = 1, 2, …, n), then the formula for calculating the total number of ants in the ant colony is as follows:

$$m = \sum_{i=1}^{n} b_i(t) \tag{7}$$

In the above formula, t represents the time; $b_i(t)$ represents the number of ants at the node position i when the time is t.

Let $\tau_{ij}(t)$ represent the amount of information participating in the connection between nodes i and j when time is t, and the information amount of all node paths is regarded as a unified value, and the movement direction of ants in the optimisation process is affected by this amount of information (Liu et al., 2022). Then when the time is t, the probability expression formula of ants moving from node i to node j is as follows:

$$p_{ij}^k = \begin{cases} \left(\left[\tau_{ij}(t)\right]^\alpha \left[\theta_{ij}\right]^\beta\right) \dfrac{1}{\sum\limits_{k \in tabu_k} \left[\tau_{ik}(t)\right]^\alpha \left[\theta_{ij}\right]^\beta}, & j \notin tabu_k \\ 0, & j \in tabu_k \end{cases} \tag{8}$$

In the above formula, α represents the importance of the heuristic information; $tabu_k$ represents the node mark passed by the ant k; θ_{ij} represents the prior value, and its expression formula is as follows:

$$\theta_{ij} = \frac{1}{d_{ij}} \tag{9}$$

After n times, the ants complete a traversal, and use the taboo table to record the movement process of the ants. At this time, the tabu table is marked as 'recorded full'. Then delete the tabu table, so that the ant colony traverses the next time (Sutar et al., 2020). After storing the current node of the ant in the ant, and calculating the paths travelled by all the ants in the ant colony, select the shortest path travelled by the ants and save it (Wang et al., 2020). As the ant colony travels more and more paths, the shortest path travelled by the ants is gradually replaced. At this time, the ant colony traversal cruise needs to be adjusted according to the information disappearance degree. The expression formula is as follows:

$$\tau_{ij}(t+1) = \tau_{ij}(t) - \rho\tau_{ij}(t) + \rho\Delta\tau_{ij} \tag{10}$$

$$\Delta t_{ij}^k = \begin{cases} \dfrac{Q}{L_k} \\ 0 \end{cases} \tag{11}$$

When the value of formula (11) is not 0, the ant k passes the path length when the time is and.

In the above formula, Δt_{ij}^{k} represents the amount of information that the kth ant stays on the path during the traversal process; L_k is the path length; Q is a variable constant.

In general, set the counter value of the ant traversal. When the number of times the ant colony traverses reaches the counter value, stop traversing. At this time, the minimum path the ant travels is the optimal solution of the node traffic model of the IoT. So far, the node traffic data between the current IoT and the user communication network has been obtained.

2.2 Traffic anomaly identification of IoT nodes based on convolutional neural network

The node traffic during the communication of the IoT has a dynamic change law of time sequence. When the node traffic is abnormal, it indicates that there is a malicious attack (Fei et al., 2020). On the basis of the temporal dynamic change law of the attacked node’s traffic, the convolutional neural network model is used to extract the characteristic information in the dynamic process of the network node’s traffic, and identify its abnormal type and abnormal node location, so as to provide a protection target for the subsequent privacy protection of the IoT node. The detailed steps are as follows:

Input IoT node traffic

Output IoT node identification results

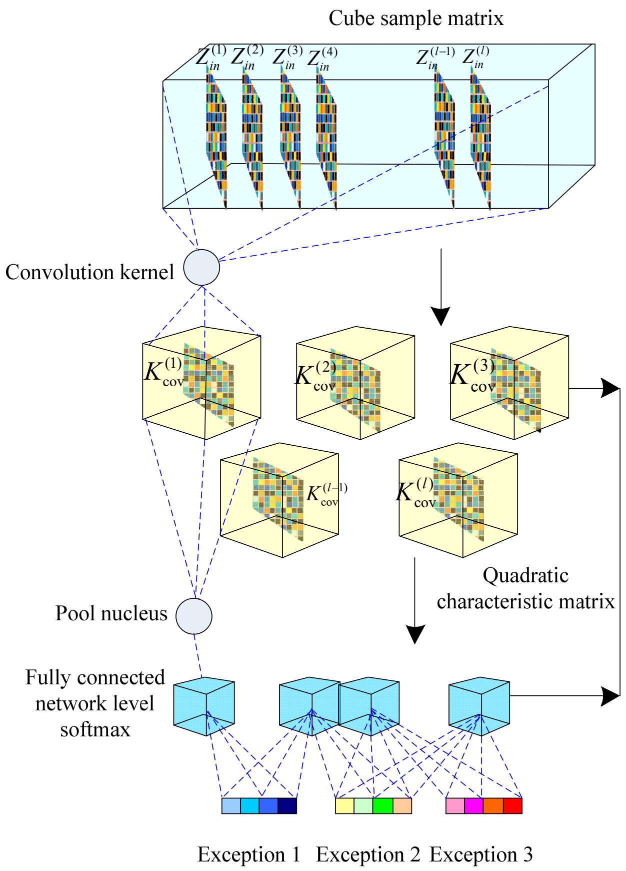
Step 1 Establish the cube sample matrix of the convolutional neural network model

Let x ∗ y ∗ t represent the three-dimensional matrix of the convolutional neural network model, where t represents the traffic timing frame of the IoT communication node, and x ∗ y can be seen as the dynamic screen of the IoT node composed of several traffic timing frames of the IoT communication node. In this way, the dynamic screen (Hassanibesheli and Donner, 2019) composed of several traffic timing frames of IoT communication nodes can be used to present the current operation process of IoT nodes. When building the convolutional neural network model, first input the IoT node flow time sequence frame into the model, normalise the time sequence frame, and then combine the IoT node flow time sequence frames at the same time to generate feature sub pixels. According to the flow time sequence frame topology of IoT nodes (Park and Demarco, 2020), arrange the flow time sequence frames of IoT nodes to generate a feature frame picture of x ∗ y. The cubic sample matrix of the convolutional neural network model can be obtained by superimposing the characteristic frame images within the time period (t_0, t_0 + 1). The matrix is represented by C_{xyt0}.

Step 2 Generate 3D feature extraction network

In order to better extract the running state characteristics of the network node traffic, it is necessary to extract its cube sample matrix from three dimensions. A three-dimensional feature extraction network of convolutional neural network is built here, and the network structure is shown in Figure 2.

Figure 2 Schematic diagram of three-dimensional feature extraction network structure of three-dimensional convolutional neural network model (see online version for colours)


Cube sample matrix
$Z_{in}^{(1)}$ $Z_{in}^{(2)}$ $Z_{in}^{(3)}$ $Z_{in}^{(4)}$ $Z_{in}^{(l-1)}$ $Z_{in}^{(l)}$
Convolution kernel
$K_{cov}^{(1)}$ $K_{cov}^{(2)}$ $K_{cov}^{(3)}$
$K_{cov}^{(l-1)}$ $K_{cov}^{(l)}$
Quadratic characteristic matrix
Pool nucleus
Fully connected network level softmax
Exception 1 Exception 2 Exception 3

The characteristic process of the network node traffic extracted from the convolutional neural network is: when the physical network node traffic is abnormal, take the cubic sample matrix of the IoT node traffic as the input, use several convolutions and start from three dimensions, and perform the sliding convolution operation on the cubic sample matrix. Let $Z_{out}^{(l)}$ and $Z_{in}^{(l)}$ represent the 3D feature volume matrix and input matrix output from the l network layer respectively, and the ⊗ symbol represents the sliding convolution operator. The formula for sliding convolution of the cube sample matrix is as follows:

$$Z_{out}^{(l)} = f\left(Z_{in}^{(l)}\right) \otimes K_{cov}^{(l)} + b^{(l)} \quad (12)$$

$$m_{xyt}^{(l)} = \left(m_{xyt}^{(l-1)} - k_{xyt} + 2p + S\right)\frac{1}{s} \quad (13)$$

In the above formula, l represents the network layer serial number; $K_{cov}^{(l)}$ represents the three-dimensional convolution kernel of layer l network; $b^{(l)}$ represents the offset value of layer l network; f(.) is the activation function; $m_{xyt}^{(l)}$ represents the three-dimensional feature volume parameters of the layer l network; k_{xyt} represents the order of convolution kernel; p and s is the structural parameter of convolutional neural network.

After the sliding convolution of the cube sample matrix is completed, the pooling check is used to pool the maximum value of the cube sample matrix, and the high-dimensional characteristics of the cube sample matrix are obtained. The output expression formula of the pooling layer is as follows:

$$Z_{out}^{(l+1)} = f_{max}\left(Z_{in}^{(l)}\right) \otimes \frac{1}{K_{sub}} \quad (14)$$

In the above formula, K_{sub} represents pooled kernel matrix; f_{max}(.)is the maximum value of the activation function.

Step 3 Identification of abnormal nodes in the IoT

Input the results of formula (14) into the full connection layer, use the hierarchical softmax classifier in this layer to classify and identify the results of formula (14), and then output the identification results of abnormal nodes of the current IoT.

Step 4 Convolution neural network model training

The training form of the convolutional neural network model is the supervised form (Sun et al., 2019), which uses the small batch gradient descent algorithm to train the model. The training process is as follows:

1 Convolutional neural network initialisation

The parameters of convolution kernel and pooling kernel of convolution neural network are initialised, and the expression formula is as follows:

$$k_{ijl} = \text{rand}\{[-1, 1]\} \quad (15)$$

In the above formula, k_{ij1} represents the parameter of convolution kernel or pooling kernel; rand{.} stands for random operation.

2 Forward propagation processing

Take a group of cube sample matrices as the input, perform convolution, pooling and full connection calculations on them, and obtain the output vector of the group of cubes. The expression formula is as follows:

$$O = f_F\left(f_S\left(f_C\left(C_{xyt}\right)\right)\right) \quad (16)$$

In the above formula, f_C(.), f_S(.) and f_F(.) represent convolution and pooled kernel full connection calculation functions respectively.

3 Back propagation

Calculate the output value of the convolutional neural network model and the error value of the training data label. The expression formula is as follows:

$$k_{xyt1} = k_{xyt0} + BpE_{12} \quad (17)$$

In the above formula, k_{xyt0} and k_{xyt1} represent the convolution kernel and pooling kernel parameters before and after adjustment respectively; Bp(.)is the gradient back propagation function of small batch; E_{12} represents the error difference between the output value of convolutional neural network model and the label error of training data.

The convolutional neural network is used to obtain the abnormal node of the unit IoT through the iterative operation of the above steps. This node is the currently attacked node, which needs privacy protection. Therefore, the next step is to build a research on the privacy protection strategy of IoT nodes.

2.3 *Privacy protection of IoT nodes*

Based on the attacked IoT nodes obtained in the above summary, the degree sequence of the attacked IoT nodes is divided by the greedy algorithm based on the average degree, and then the graph structure modification method is used to realise the privacy protection of the IoT nodes. The detailed process of dividing the degree sequence of IoT nodes using the greedy algorithm based on average degree is as follows:

The entire IoT is described using an undirected and unweighted graph (Yamin et al., 2019), which is represented by G = (V, E), where V = {v_1, v_2, …, v_n} represents the set of attacked user entity nodes, and E = {(v_i, v_j)|v_i, v_j ∈ V} represents the set of edges between nodes, the association between user nodes. The degree sequence of the IoT undirected unweighted graph is denoted by Y_G, and Y = {y_1, …, y_n}, where d_i represents the degree of the i node.

Taking k attacked IoT nodes and IoT undirected and unweighted graph as input, use sorting algorithm to sort the degree sequence of IoT undirected and unweighted graph, and save k attacked IoT nodes to within a group. Then calculate the cost of whether the k + 1 attacked IoT node is stored in this group or whether it is a new group. The calculation formula is as follows:

$$C_{merge} = \left|y(a) - y(k+1)\right| + W(y[k+2, 2k+1]) \tag{18}$$

$$C_{new} = W(y[k+1, 2k]) \tag{19}$$

In the above formula, C_{new} represents the cost of depositing the k + 1 attacked network node into the new group; C_{merge} represents the cost of depositing the k + 1 attacked network node into the existing group; I(y[k + 2, 2k + 1]) represents the k + 1 attacked network node k-anonymity cost; W represents the total number of nodes.

Compare the result of formula (18) with the result of formula (19), when the result of formula (18) is greater than the result of formula (19), the k + 1 to 2k nodes are stored in the new group, otherwise, they are combined and stored within the previous group. When all attacked IoT nodes are grouped, calculate the average degree value of all attacked network nodes. The expression formula is as follows:

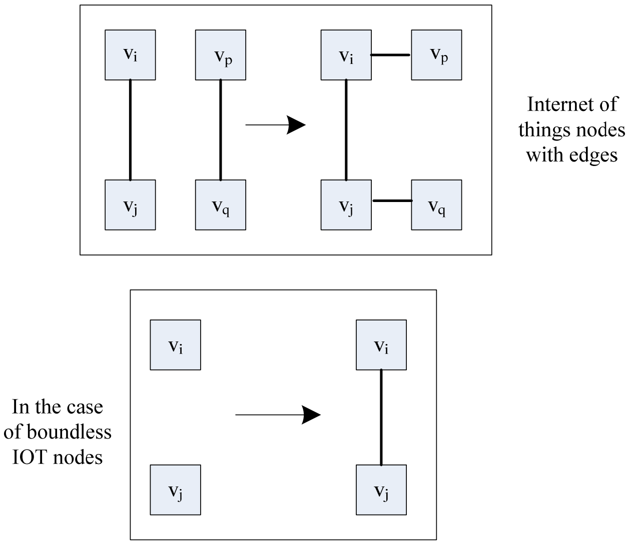
$$y(a) = \frac{\sum_{l=i}^{j} y(W)}{j-i+1} \tag{20}$$

In the above formula, i and j both represent IoT nodes; y(W) represents the degree sequence of W nodes.

After calculating the average degree value of all attacked IoT nodes using formula (20), replace all node degrees in all groups with the average degree value, and thus obtain the k-degree anonymous sequence of attacked IoT nodes.

Using the k-degree anonymous sequence of the attacked IoT nodes, the undirected and unweighted graph of the IoT is modified using the edge addition strategy, as shown in Figure 3.

Figure 3 Schematic diagram of adding strategy to modify the undirected and unauthorised graph of the IoT (see online version for colours)


v_i
v_p
v_i
v_p
Internet of things nodes with edges
v_j
v_q
v_j
v_q
v_i
v_i
In the case of boundless IOT nodes
v_j
v_j

When IoT nodes v_i and v_j are edgeless, determine whether the two nodes belong to a k-degree anonymous sequence, if so, add an edge between the two IoT nodes, and calculate the degree of the two nodes Also add 1. When there is an edge between IoT nodes v_i and v_j, find two nodes v_p and v_q in their adjacent node groups, and judge whether nodes v_p and v_q belong to a k-degree anonymous sequence, if so, then Delete the edge between node v_p and v_q. Connect nodes v_i with v_p, v_j and v_q, and add new edges. So far, the modification of the undirected and unauthorised graph of the IoT has been completed, and the privacy protection of the attacked IoT nodes has been realised.

The above process uses the ant colony algorithm to solve the constructed IoT network node traffic model, and obtains the traffic data of the physical network node. Based on the obtained traffic data, a convolutional neural network model is constructed, and the identification of abnormal nodes in the IoT is completed through iterative training and calculation. According to the identification results of the identified IoT abnormal nodes, the privacy protection strategy of k-anonymous IoT nodes based on the average degree of nodes is adopted to protect the privacy of IoT abnormal nodes, so as to improve the security of IoT nodes.

3 Experiment analysis

Taking the IoT built in a logistics park as the experimental object, the method of this paper is used to protect the privacy of its nodes. During a certain period of time, the traffic data of the IoT node is collected, including 8,043 network nodes, and 168,903 undirected edges in the 8,043 IoT nodes. The average degree of all IoT nodes is 46 degrees, and the node degree is based on Power-law distribution. Take 80% of the collected data as the training sample and 20% of the data as the test sample. The IoT structure of the logistics park is shown in Figure 4.

3.1 IoT node traffic collection test

Taking the node traffic of the IoT server as the experimental object, the method of this paper is used to collect the node traffic of the server in a certain period of time, and the capacity of the IoT node traffic collection of the method in this paper is analysed. The results are shown in Figure 5.

Analysis of Figure 5 shows that when the IoT node is attacked, the IoT node traffic shows a rapid upward trend, and the node traffic fluctuates slightly. When the IoT node is not attacked, the traffic value is small and stable. However, when the method in this paper collects the traffic of IoT nodes, the values before and after the network attack are exactly the same as the actual values. In the process of network attack, there is a slight deviation between the IoT node traffic collected by this method and its actual value, but the deviation value is only about 2 kb/s, and the IoT node traffic curve collected by this method when attacked is very consistent with its actual curve. The above results show that the method in this paper can more accurately collect the current IoT node traffic when the network is attacked and secure, and provide a better foundation for the privacy protection of subsequent IoT nodes.

Figure 4 Schematic diagram of IoT structure of logistics park (see online version for colours)

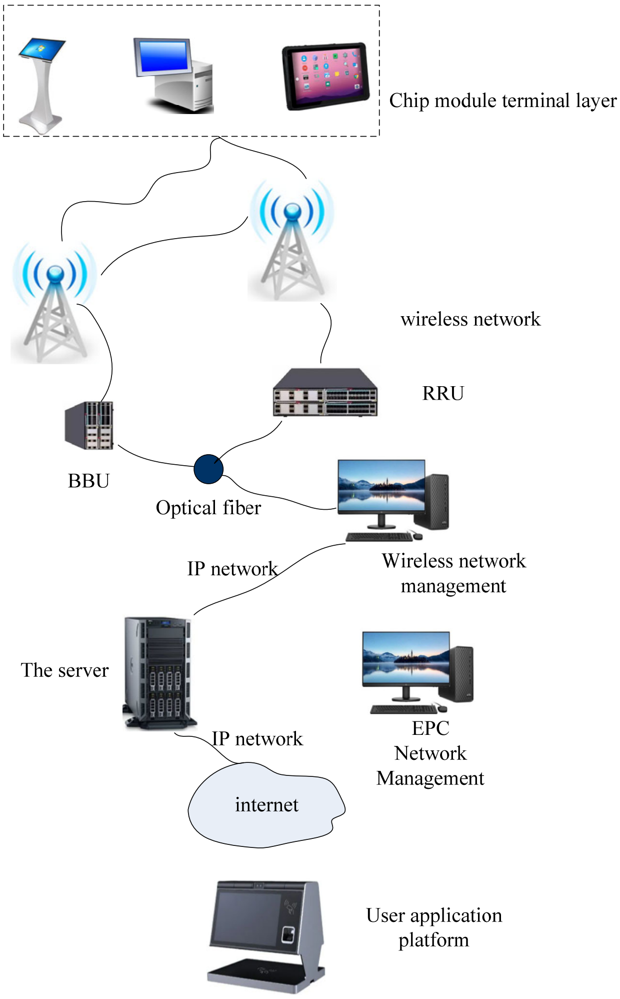
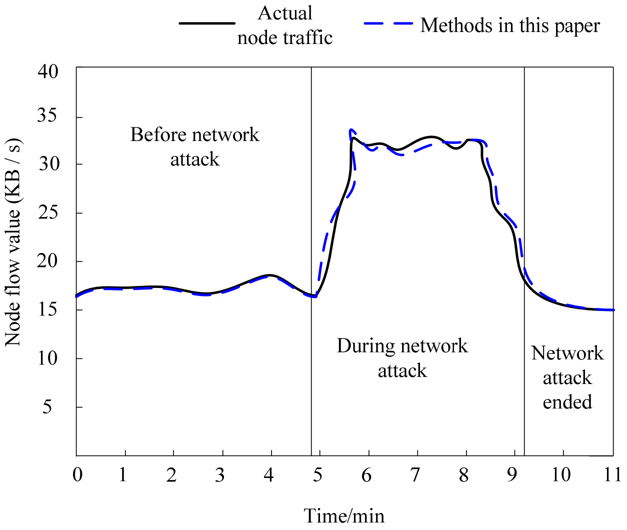


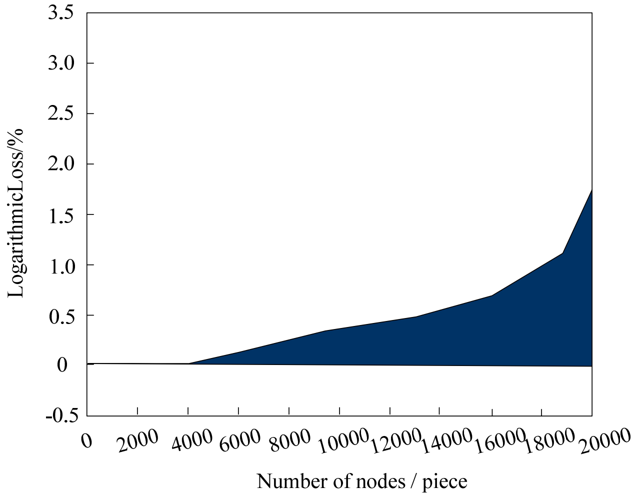
Figure 5 Traffic collection test results of IoT nodes (see online version for colours)



3.2 IoT node anomaly identification test

The logarithmic loss is used as the measurement index, and the logarithmic loss threshold is set to 3.0% to test the performance of the algorithm in this paper to identify different abnormal IoT nodes under attack under different data volumes of IoT nodes. The results are shown in Figure 6.

Figure 6 Change of logarithm loss when identifying IoT abnormal nodes (see online version for colours)



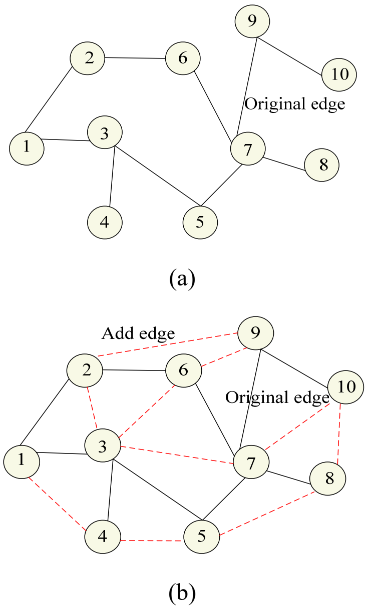
Analysis of Figure 6 shows that the logarithmic loss value of the method in this paper when identifying abnormal nodes in the IoT is proportional to the number of nodes. However, when the number of IoT nodes is before 4,000, the logarithmic loss value of the method in this paper to identify abnormal IoT nodes is always 0. When the number of IoT nodes exceeds 4,000, the logarithmic loss value of the method used in this paper to identify abnormal IoT nodes shows a slow upward trend. When the number of IoT nodes exceeds 16,000, the logarithmic loss value when identifying abnormal IoT nodes increases. The magnitude increased slightly. However, the maximum logarithmic loss value of the method in this paper to identify abnormal nodes of the IoT is only about 1.6%, which is far lower than the set logarithmic loss value threshold. The above results show that the logarithmic loss value of the method in this paper is small when identifying abnormal nodes in the IoT, and the identification results are more accurate.

3.3 Node privacy protection test

Taking a certain IoT attacked abnormal node group as the experimental object, there are 10 abnormal nodes in this group, the privacy of this abnormal node is protected by the method in this paper, and the protection result is shown in Figure 7.

Analysis of Figure 7 shows that there are ten edges between the initial abnormal nodes of the IoT, and after using the method in this paper to protect the node privacy, ten new edges are added on the basis of the original edges, so that the current IoT node There are 20 edges in total. This situation changes the undirected and unauthorised graph of the IoT, so that the privacy of IoT nodes is better protected.

Figure 7 IoT abnormal node edge after node privacy protection, (a) initial node edge (b) add edge (see online version for colours)



Taking the amount of information loss as a measure, the method in this paper is tested to protect the changes in the amount of information loss of different IoT nodes and the node under different attack degrees. The results are shown in Table 1.

Table 1 change of information loss amount of IoT privacy protection (%)

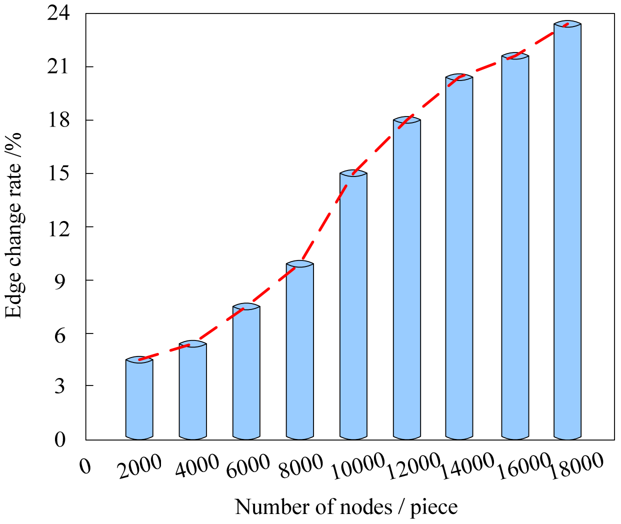
Number of IoT nodes (piece)	*Attack risk level*		
	Low	*Moderate*	*High*
500	2.25	2.26	2.28
1,000	2.25	2.26	2.33
1,500	2.25	2.28	2.39
2,000	2.26	2.29	2.42
2,500	2.27	2.31	2.46
3,000	2.29	2.34	2.51
3,500	2.31	2.39	2.59
4,000	2.34	2.46	2.63
4,500	2.38	2.51	2.66
5,000	2.41	2.59	2.73
5,500	2.46	2.63	2.78
6,000	2.53	2.68	2.81
6,500	2.61	2.72	2.84
7,000	2.77	2.76	2.89

Analysis of Table 1 shows that the amount of information loss when protecting the privacy of IoT nodes by this method increases with the increase of IoT privacy nodes, and when the risk of IoT attacks is low, the method in this paper protects the information loss of IoT node privacy. The amount is slightly lower. When the number of IoT nodes is before 3,000, the information loss when the method of this paper protects the privacy of IoT nodes shows a small increase trend at different attack risk levels. When the number of IoT nodes exceeds 3,000, the amount of information loss when protecting the privacy of IoT nodes by the method in this paper increases slightly with different attack risk levels. Among them, when the attack risk is high and the number of IoT nodes is 7,000, the information loss value of this method to protect the privacy of IoT nodes is 2.89%, which is 3.11% lower than the set threshold. The above results show that when the method in this paper protects the privacy of IoT nodes, it is less affected by the number of nodes and the degree of attack on the IoT, and its information loss value is low.

The method in this paper is tested with the edge change rate of the IoT node as a measure.

The practical application effect of this method is further verified. Taking the rate of change of the edge when protecting the privacy node of the IoT as a measure, the rate of change of the edge of the method in this paper is tested in the case of different number of privacy protection nodes of the IoT, and the results are shown in Figure 8.

Figure 8 Edge change rate of IoT node privacy protection (see online version for colours)



It can be seen from the analysis of Figure 8 that the edge change rate when protecting the privacy of IoT nodes in this method is in direct proportion to the number of IoT nodes, that is, the more IoT nodes, the higher the edge change rate when protecting their privacy. When the number of IoT nodes is 18,000, the edge change rate of this method is close to 24% after protection. The results show that the method in this paper can effectively adjust the relationship between IoT nodes, add edges between IoT nodes, make the change rate of IoT node edges larger, and increase its privacy security.

The method in this paper is verified from the point of view of the average degree of nodes in the process of privacy protection of IoT nodes. When testing the method in this paper to protect the privacy of IoT nodes, the change of the average degree of nodes in the undirected and unauthorised graph with different numbers of original network nodes is protected. The results are shown in Table 2.

Table 2 Average degree value of IoT nodes during privacy protection (%)

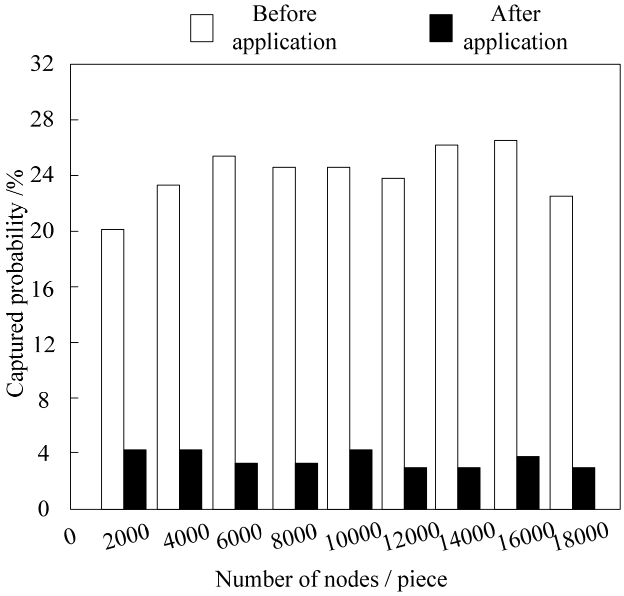
Number of IoT nodes (piece)	*Original undirected graph*	*Undirected and unauthorised diagram after protection*
500	42.4	55.8
1,000	42.5	59.1
1,500	42.8	62.5
2,000	43.1	68.9
2,500	43.9	70.1
3,000	44.4	71.6
3,500	44.8	73.5
4,000	45.1	74.7
4,500	45.3	75.5
5,000	45.9	75.9
5,500	46.5	76.8
6,000	46.9	77.2
6,500	47.3	77.9
7,000	47.8	78.4

Table 2 shows that the average degree of IoT nodes is positively correlated with the number of nodes. Among them, in the case of the same number of IoT nodes, the average degree value of the original IoT undirected and unauthorised nodes in the graph shows a small increase trend. After the privacy of this IoT node is protected in this paper, the average degree value of the protected IoT undirected and unauthorised nodes in the graph shows a large increase trend. When the number of IoT nodes is 7,000, the average degree of nodes in the undirected powerless graph of IoT protected by this method is 30.6% higher than that in the original undirected powerless graph of IoT. The above results show that after the application of this method, the average value of IoT nodes can be effectively improved, the security between nodes can be increased, and the privacy protection ability of IoT nodes is better.

Taking the privacy capture probability of the protected IoT node as a measure, test the capture probability of the privacy of the IoT node before and after being protected under different IoT node numbers. The results are shown in Figure 9.

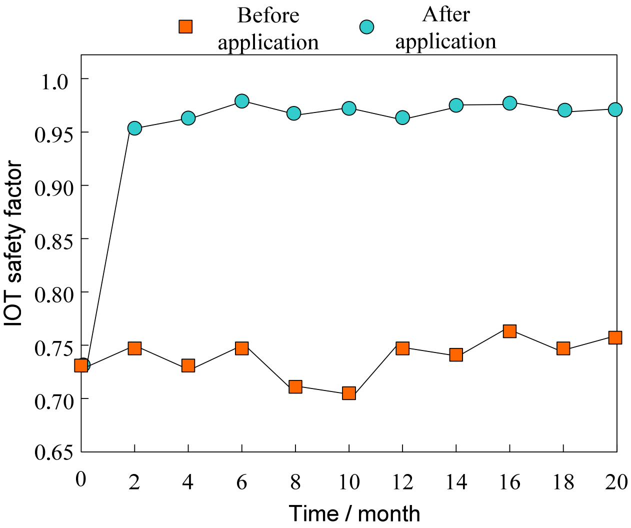
It can be seen from the analysis of Figure 9 that when the IoT nodes are different, the probability of node privacy capture is irregularly distributed. The reason is that there are many kinds of network attacks, and the captured network nodes are also different. Therefore, the probability of node privacy capture cannot be analysed from the perspective of the number of network nodes. However, in the case of the same number of network nodes, after the privacy protection of the node in this method, the probability of the network node failing is low. Before the application of this method, the probability of privacy capture of the IoT node is between 20% and 28%, and after the application, the probability of privacy capture of the network node is about 4%. The results show that this method can effectively reduce the probability of privacy capture of IoT nodes.

Figure 9 Probability value of privacy capture of IoT nodes



Taking the safety factor of the IoT as the measurement index, test the change of the safety factor of the IoT before and after the application of this method, and the results are shown in Figure 10.

Figure 10 IoT safety factor before and after application (see online version for colours)



It can be seen from the analysis of Figure 10 that the security factor of the IoT fluctuates both before and after the application of this article. This is because the IoT is subject to attacks from different sources and types every day. After the application of the method in this paper, the safety factor of the IoT presents a straight upward trend. Two months after the application of this method, the safety factor of the IoT has reached 0.95. With the increase of time, the safety factor of the IoT shows a small fluctuation trend, which is far lower than the fluctuation value of the safety factor of the IoT before the application. To sum up, the method in this paper can effectively improve the security factor of the IoT, and its privacy protection ability of the IoT node is relatively significant.

4 Conclusions

As the carrier of data exchange, the IoT node aims to improve the security of IoT data exchange, and proposes a privacy protection method for IoT nodes based on convolutional neural network algorithm. The convolutional neural network model is applied to identify abnormal nodes in the current IoT, which is the key to realise the privacy protection of nodes in the IoT. According to the identified abnormal nodes of the IoT, k-anonymous IoT node privacy protection strategy based on the average degree of nodes is adopted to protect the privacy of abnormal nodes of the IoT. The experimental results show that the method in this paper has a better ability to collect the traffic of IoT nodes and a more accurate ability to identify abnormal IoT nodes in the application process. The deviation value is only 2kb/s. It can also effectively protect the privacy of IoT nodes and improve their security coefficient. It has a good application effect.

References

Bica, I., Chifor, B.C., Arseni, T.C. and Matei, I. (2019) 'Multi-layer IoT security framework for ambient intelligence environments', *Sensors*, Vol. 19, No. 18, pp.4038–4042.

Dimitriou, K. and Roussaki, I. (2019) 'Location privacy protection in distributed IoT environments based on dynamic sensor node clustering', *Sensors*, Vol. 19, No. 13, pp.3022–3029.

Fei, J., Yao, Q., Chen, M., Wang, X. and Fan, J. (2020) 'The abnormal detection for network traffic of power IoT based on device portrait', *Scientific Programming*, Vol. 20, No. 9, pp.1–9.

Gao, J., Yi, S., Bing, Z., Chen, Z. and Gao, L. (2019) 'Multi-GPU based parallel design of the ant colony optimization algorithm for endmember extraction from hyperspectral images', *Sensors*, Vol. 19, No. 3, pp.598–604.

Hassanibesheli, F. and Donner, R.V. (2019) 'Network inference from the timing of events in coupled dynamical systems', *Chaos*, Vol. 29, No. 8, pp.83–89.

Lakshmanaprabu, S.K., Shankar, K., Rani, S.S., Abdulhay, E., Arunkumar, N. and Ramirez, G. (2019) 'An effect of big data technology with ant colony optimization based routing in vehicular ad hoc networks: towards smart cities', *Journal of Cleaner Production*, Vol. 217, No. 20, pp.584–593.

Li, F., Liu, M. and Xu, G. (2019a) 'A quantum ant colony multi-objective routing algorithm in WSN and its application in a manufacturing environment', *Sensors*, Vol. 19, No. 15, pp.3334–3339.

Li, S., Liu, Z., Huang, Z., Lyu, H. and Liu, W. (2019b) 'Dynapro: dynamic wireless sensor network data protection algorithm in IoT via differential privacy', *IEEE Access*, Vol. 32, No. 9, pp.103–108.

Li, X. and Wu, J. (2020) 'Node-oriented secure data transmission algorithm based on IoT system in social networks', *IEEE Communications Letters*, Vol. 17, No. 2, pp.12–19.

Liu, C., Shen, W., Zhang, L., Du, Y. and Yuan, Z. (2021) 'Spike neural network learning algorithm based on an evolutionary membrane algorithm', *IEEE Access*, Vol. 48, No. 8, pp.19–25.

Liu, Y., You, X. and Liu, S. (2022) 'Multi-ant colony optimization algorithm based on hybrid recommendation mechanism', *Applied Intelligence*, Vol. 52, No. 8, pp.8386–8411.

Meng, X., Nie, L. and Song, J. (2019) 'IoT individual privacy features analysis based on convolutional neural network', *Cognitive Systems Research*, Vol. 57, No. 12, pp.126–130.

Park, B. and Demarco, C.L. (2020) 'Optimal network topology for node-breaker representations with ac power flow constraints', *IEEE Access*, Vol. 36, No. 5, pp.21–26.

Park, H. (2020) 'Anti-malicious attack algorithm for low-power wake-up radio protocol', *IEEE Access*, Vol. 22, No. 99, pp.149–153.

Sun, L., Huang, N., Wang, L., Wang, Q. G. and Zhang, Y. (2019) 'A network application model with operational process feature', *Journal of the Franklin Institute*, Vol. 356, No. 12, pp.6678–6696.

Sutar, S.G., Mali, P.J. and More, A.Y. (2020) 'Resource utilization enhancement through live virtual machine migration in cloud using ant colony optimization algorithm', *International Journal of Speech Technology*, Vol. 23, No. 1, pp.79–85.

Taguchi, S. and Yoshimura, T. (2021) 'Online estimation and prediction of large-scale network traffic from sparse probe vehicle data', *IEEE Transactions on Intelligent Transportation Systems*, Vol. 63, No. 18, pp.1–11.

Wang, W., Cai, Z., Zhao, J. and Si, S. (2020) 'Optimization of linear consecutive- k -out-of- n systems with Birnbaum importance based ant colony optimization algorithm', *Journal of Shanghai Jiaotong University (Science)*, Vol. 25, No. 2, pp.253–260.

Yamin, M., Alsaawy, Y., Alkhodre, A.B. and Sen, A. (2019) 'An innovative method for preserving privacy in internet of things', *Sensors*, Vol. 19, No. 15, pp.335–341.

Zhang, P., Wang, Y., Kumar, N., Jiang, C. and Shi, G. (2021) 'A security and privacy-preserving approach based on data disturbance for collaborative edge computing in social IoT systems', *IEEE Transactions on Computational Social Systems*, Vol. 49, No. 18, pp.1–12.