Computers and Electrical Engineering 77 (2019) 339–353

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Detecting malicious nodes via gradient descent and support vector machine in Internet of Things ☆

Liang Liu [a], Jingxiu Yang [a], Weizhi Meng [b,*]

[a] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, 211106 Nanjing, China
[b] Department of Applied Mathematics and Computer Science, Technical University of Denmark, Denmark

ARTICLE INFO

Article history:
Received 29 January 2019
Revised 21 May 2019
Accepted 17 June 2019
Available online 27 June 2019

Keywords:
Internet of things
Malicious node detection
Support vector machine
Gradient descent
K-means
Trust management
Machine learning

ABSTRACT

IoT devices have become much popular in our daily lives, while attackers often invade network nodes to launch various attacks. In this work, we focus on the detection of insider attacks in IoT networks. Most existing algorithms calculate the reputation of all nodes based on the routing path. However, they rely heavily on the assumption that different nodes in the same routing path have equal reputation, which may be not invalid in practice and cause inaccurate detection results. To solve this issue, we formulate it as a multivariate multiple linear regression problem and use the *K*-means classification algorithm to detect malicious nodes. Further, we optimize the routing path and design an enhanced detection scheme. Our results indicate that our proposed methods could achieve a detection accuracy rate of 90% or above in a common case, and the enhanced scheme could reach an even lower false detection rate, i.e., below 5%.

1. Introduction

Currently, IoT devices are becoming more and more popular in our daily lives, and have broad application prospects in many fields such as smart home, smart health, public safety, industrial monitoring and environmental protection [1]. Various IoT devices are interconnected to exchange information with each other. The sensory data are usually forwarded by multiple devices in a multi-hop ad hoc network and are sent to the aggregation node to provide data support for user decision-making. The mesh topology has been adopted in many IoT protocols, such as Z-Wave, Thread and ZigBee/IEEE 802.15.4 [2]. It allows devices having flexibility to communicate with other devices within its communication range, as well as with sink through multi-hop routing.

IoT is developing very rapidly, but IoT devices are facing many security challenges. Considering various attacks in IoT networks, IoT devices could be vulnerable to many attacks: (i) Passive attack: an attacker can eavesdrop on communication between nodes through a wireless channel to obtain valuable information. Since passive attack only steals information without compromising the normal operation of the protocol, it is very difficult to detect. (ii) Active attack (e.g., Black hole attack [3], Sybil Attack [4]): a malicious node actively destroys network protocols and violates the security policy by injecting false information, dropping and modifying data packets, which can directly affect network availability and security. According to the sources of active attacks, they can be divided into external attack and internal attack. The external attack is a kind of

☆ This paper is for CAEE special section SI-aisec. Reviews processed and recommended for publication to the editor-in-chief by guest editor Dr. James Park.

* Corresponding author.
E-mail addresses: liangliu@nuaa.edu.cn (L. Liu), yangjingxiu613zb@163.com (J. Yang), weme@dtu.dk (W. Meng).

https://doi.org/10.1016/j.compeleceng.2019.06.013
0045-7906/

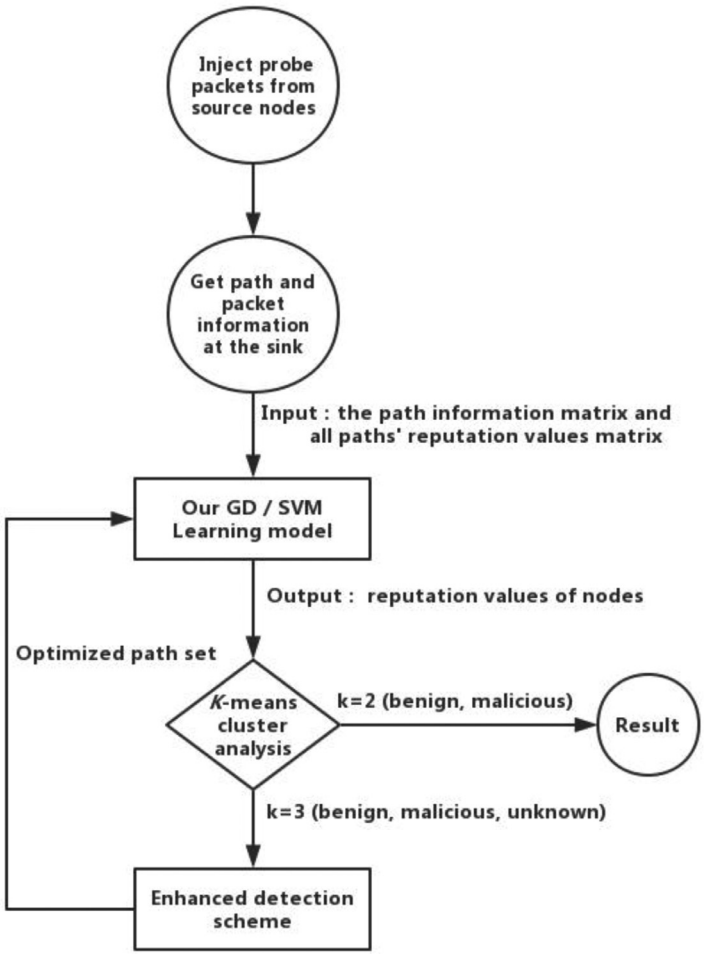

Fig. 1. Workflow of our proposed detection approach.

attack on the network by a node outside the network that has not been verified; while the internal attack is a kind of attack that some nodes are invaded or malicious within the network, i.e., have the right to access network resources.

Malicious nodes can cause enormous damage to the network and system functions; therefore, it is very critical to identify malicious nodes timely. In the literature, many studies focused on detecting and identifying malicious behavior and traffic. Targeting on this issue, in this paper, we formalize the relationship between the reputation of routing paths and nodes, by observing that a node's reputation can be formalized as a multiple linear regression problem. Then we use the gradient descent (GD) algorithm and the support vector machine (SVM) algorithm to learn the node's trust value and detect malicious nodes. Both algorithms can adjust the detection model to obtain the accurate node's reputation according to the input information.

As shown in Fig. 1, we show the workflow of our proposed approach. We first inject probe packets from trusted source nodes into the network. The sink then analyzes the received packets and extracts all transmission paths & trust values of the paths. Then, we use GD and SVM algorithm to learn the reputation values of all nodes and cluster the nodes into three groups, such as benign group (BG), unknown group (UG) and malicious group (MG). To further enhance the detection, we optimize network routing paths and re-inject the packets to collect information about those nodes in UG. We train our learning model again using the information collected from the sink and output the final results, i.e., clustering all nodes into two groups, like the final benign group (FBG) and the final malicious group (FMG). In our evaluation, our results show that under the enhanced detection scheme, our proposed method could achieve higher detection accuracy and lower false detection rate. The contributions can be summarized as follows.

- We first formulate the detection of malicious nodes as a multivariate multiple linear regression problem. Then we propose a method of learning a node's reputation by using both the gradient descent (GD) algorithm and the support vector machine (SVM) algorithm.
- We use the *K*-means clustering algorithm to detect the malicious nodes. Also, we optimize the routing path and design an enhanced detection scheme to further improve the detection performance.

- The experimental results show that our approach could achieve better detection accuracy than similar work, i.e., the detection accuracy of our approach is above 90%.

The remainder of the paper is organized as follows: Section 2 introduces related work on detection of malicious nodes in IoT networks. Section 3 describes our proposed network model and detection approach. Section 4 presents the experimental results, and Section 5 concludes the paper.

2. Related work

How to detect malicious nodes in the IoT network is a challenge. In [5, 6], network diversity was used to identify malicious relay nodes in a mesh network. However, the scheme incurs a lot of overhead, and high-precision identification can only be guaranteed based on the assumption that there is at least one reliable path between the source and the sink. In [7], a novel solution was proposed, called Ad Hoc On-Demand Distance Vector (AODV) black hole detection and removal (AODV-BDR), which can detect and remove black hole attack by using the neighbor-based authentication technique.

In recent years, there are many research studies on the detection of malicious nodes in IoT based on credibility or trust models. In [8], Rikli and Alnasser proposed a lightweight trust-based model in which each node monitors its one-hop neighbor's behavior and information, which is ultimately collected by the base station and is evaluated under some trust measures. The trust measures are based on current and past trust values to detect any malicious nodes. In [9], Chen et al. proposed a trust evaluation model and data fusion mechanism. In the trust evaluation model, trust evaluation consists of three parts: behavior trust, data trust, and historical trust. The authors analyzed the performance of a developed trust determination algorithm, namely, the Neighbor-Weight Trust Determination (NWTD) algorithm in [10], which is based on weighted voting. A node receives different trust values for the same one-hop neighbor from different nodes. They then used the weighted average method for calculating the final trust value [9, 10]. In [11], they used two metrics to help detect untruthful information and malicious nodes: (a) the similarity of recommendations from different sources, and (b) the consistency of the received information from a particular source. A trust management scheme was proposed in [12], which leveraged the Dempster-Shafer evidence theory. By taking into account spatiotemporal correlation of the data collected by sensor nodes in adjacent area, the trust degree can be estimated. In [13], Liu et al. proposed a hierarchical network architecture that utilized multiple link communication technologies (dual link technology and single-link technology) with different characteristics to infer the node reliability for identifying malicious nodes. However, their work was based on the assumption that DL nodes are reliable and trusted. Then, a weighted trust-based malicious node detection scheme was proposed in [14], in which the sensor readings of neighboring nodes of each sensor node are reported to a cluster head for data aggregation. Then they calculated the trustworthiness of the corresponding sensor readings, as weights in the weighted majority voting method to help detect malicious nodes.

On the other hand, machine learning methods are widely used to help identify cyber-attacks and malicious nodes. In [15, 16], Support Vector Machine (SVM) was used to detect attacks, but it is difficult to ensure that each node is one-hop away from a trusted node in the multi-hop network. Both studies [17, 18] applied unsupervised machine learning algorithms to detect anomalies. All of them focused on identifying anomalous traffic in the network, but did not identify the attackers. In [19], the paper applied auto encoder neural networks into Wireless Sensors Networks (WSN) to identify malicious nodes. In [20], Ayadi et al. proposed a leakage detection model based on the Fisher Discriminant Analysis (FDA) and the SVM classifier, which could identify outliers based on WSN in a water pipeline. Later, in [21], Liu et al. proposed a reputation metric to measure each routing path, and then used the unsupervised learning to identify malicious nodes in a multi-hop IoT network. However, their scheme was based on the assumption that the reputation values of different nodes in the same routing path are equal to each other, which may be not realistic in practice and result in inaccurate results.

In this paper, we relax such assumption and consider a general attack model. Based on the reputation metric, we formulate the task of detecting malicious nodes as a multivariate multiple linear regression problem, and use the GD algorithm and SVM algorithm to calculate the reputation of all nodes. We further design an enhanced detection (ED) scheme to improve the performance of our detection method.

3. Our proposed network model and detection approach

According to the method in [21], we can inject the probe packets from the trusted sources. Then probe packets transmitted in the network can use data provenance [22] to record the transmission path without any support from the facilities, and can extract its complete path information by analyzing the sink. The sink can use a hash function to check the packet's integrity and obtain the reputation of a path. Then, the sink learns the trust values of nodes based on the path's information. In the end, the sink can use the clustering algorithm to identify malicious nodes. As shown in Fig. 2, suppose S is the trusted source and D is the sink. When S sends a packet, D can receive a copy of the packet from each available path. We assume that malicious nodes can launch an attack with a fixed probability P_i, then $\overline{P_i} = 1 - P_i$ indicates the probability that the node forwards unmodified packets. If the node is benign, then $P_i = 0$.

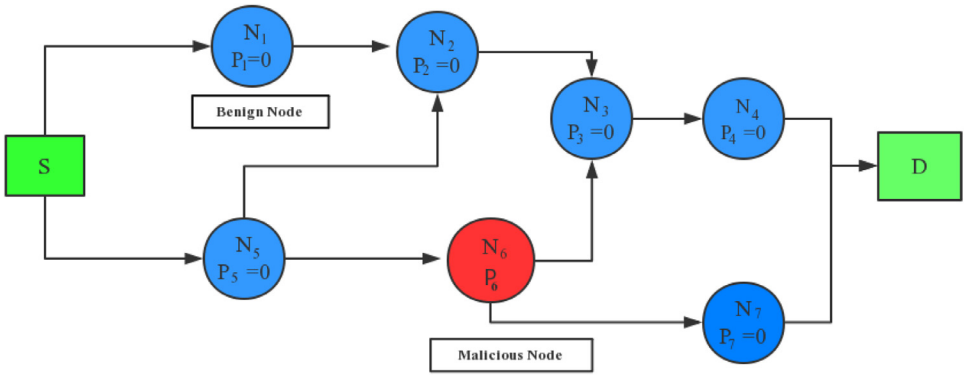

Fig. 2. An example of mesh network, *S* is the trusted source and *D* is the sink. Benign nodes are in blue and malicious nodes are in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.1. Reputation of routing paths and nodes

To start, *S* sends multiple probe packets to *D* through all possible paths. Then *D* receives all probe packets and checks the integrity of each received probe packet by using a keyed hash function. Based on the information provided by *D*, the sink can obtain the proportion of the unmodified packets along the path. The higher the proportion, the higher reputation of the corresponding path.

Let *N.T* denote the node's reputation, we can have:

$$N.T = 1 - P\,(P \text{ is the probability of launching an attack by node } N) \tag{1}$$

We further define the *Pa.T* as trust value of a routing path *Pa*, which equals to the number of unmodified packets divided by the number of all packets transmitted through path *Pa*. The reputation of path *Pa* also indicates to what degree each node within a routing path contributes to the unmodified packets. Taking Fig. 2 as an example, we can have the reputation for a path $Pa = < N_1\text{–}N_2\text{-}\ N_3\text{–}N_4>$ as below:

$$Pa.T = N_1.T * N_2.T * N_3.T * N_4.T = \frac{\text{the number of unmodified packets transmitted through path } Pa}{\text{the number of all packets transmitted through path } Pa} \tag{2}$$

3.2. Problem definition and formalization

Let Pa_i(i∈[1, *m*]) denote the *i*th routing path between *S* and *D*. Also, each Pa_i can be expressed as a set of information about all nodes. For example, as shown in Fig. 2, the path $\langle N_1\text{–}N_2\text{–}N_3\text{–}N_4\rangle$ can be represented as $Pa_i = \{1, 1, 1, 1, 0, 0, 0\}$, in which 0 and 1 indicate whether the node is contained in the path Pa_i.

Assume that there are *n* nodes in the network; then the set of *m* paths from *S* to *D* can be expressed as follows:

$$\begin{aligned} Pa_1 &= \{a_{11}\, a_{12}\, a_{13} \ldots a_{1n}\} \\ Pa_2 &= \{a_{21}\, a_{22}\, a_{23} \ldots a_{2n}\} \\ Pa_3 &= \{a_{31}\, a_{32}\, a_{33} \ldots a_{3n}\} \\ &\cdots \\ Pa_m &= \{a_{m1}\, a_{m2}\, a_{m3} \ldots a_{mn}\} \end{aligned} \tag{3}$$

$$a_{ij} = \begin{cases} 0, & If\ node\ N_j\ is\ not\ in\ path\ i \\ 1, & If\ node\ N_j\ is\ in\ path\ i \end{cases} \tag{4}$$

The reputation of the routing path *Pa* can be changed to:

$$\begin{aligned} \ln Pa.T &= \ln N_1.T + \ln N_2.T + \ln N_3.T + \ln N_4.T \\ &= \ln \frac{\text{the number of unmodified packets transmitted through the path } Pa}{\text{the number of all packets transmitted through the path } Pa} \end{aligned} \tag{5}$$

Table 1
Evaluation metrics.

Detection precision (P_d)	Ratio of malicious nodes correctly identified in the nodes set of detection results
Detection accuracy (A_d)	Ratio of malicious nodes correctly identified in the real malicious set
False positive rate (F_d)	Ratio of benign nodes misidentified in the benign nodes set

Based on the above formula, we can formalize the relationship between the reputation of all routing paths and the reputation of all nodes. For the set of m paths, we can follow Eq. (5), while for all paths, we can have the followings:

$$\begin{aligned}
&\ln Pa_1.T = a_{11} * \ln N_1.T + a_{12} * \ln N_2.T + \ a_{13} * \ln N_3.T + \ldots + a_{1n} * \ln N_n.T \\
&\ln Pa_2.T = a_{21} * \ln N_1.T + a_{22} * \ln N_2.T + a_{23} * \ln N_3.T + \ldots + a_{2n} * \ln N_n.T \\
&\ln Pa_3.T = a_{31} * \ln N_1.T + a_{32} * \ln N_2.T + a_{33} * \ln N_3.T + \ldots + a_{3n} * \ln N_n.T \\
&\qquad\qquad \ldots \\
&\ln Pa_m.T = a_{m1} * \ln N_1.T + a_{m2} * \ln N_2.T + a_{m3} * \ln N_3.T + \ldots + a_{mn} * \ln N_n.T
\end{aligned} \tag{6}$$

Then we can construct a matrix *NTM* (Node Trust Matrix) to represent the trust values for all nodes:

$$NTM = (\ln (N_1.T), \ln (N_2.T), \ln (N_3.T), \ldots, \ln (N_n.T)) \tag{7}$$

We can also have a matrix *PM* (Path Matrix) to represent the information of all paths:

$$PM = \ (Pa_1, Pa_2, Pa_3 \ldots Pa_m) = \begin{pmatrix} a_{11} & a_{21} & a_{31} & \ldots & a_{m1} \\ a_{12} & a_{22} & a_{32} & \ldots & a_{m2} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{1n} & a_{2n} & a_{3n} & \ldots & a_{mn} \end{pmatrix} \tag{8}$$

where

$$a_{ij} = \begin{cases} 0, & If\ node\ N_j\ is\ not\ in\ path\ i \\ 1, & If\ \ node\ N_j\ is\ in\ path\ i \end{cases} \tag{9}$$

We can also construct a matrix *PTM* (Path Trust Matrix) based on the reputation of all paths:

$$PTM = (ln(Pa_1.T), ln(Pa_2.T), ln(Pa_3.T), \ldots, ln(Pa_{\mathrm{m}}.T)) \tag{10}$$

According to (6), we can have:

$$NTM * PM = PTM \tag{11}$$

After we analyze the packets at the sink, *PM* and *PTM* can be known but our goal is to estimate *NTM* accurately. The more accurate a node's trust value we obtained, the more precise we can identify malicious nodes. We found that estimating *NTM* is a multiple linear regression problem, and that GD algorithm and SVM algorithm can be used to solve this problem effectively.

3.3. GD and SVM algorithm

The gradient descent (GD) algorithm and the support vector machine (SVM) algorithm are both optimization algorithms, which are commonly used for solving linear regression problems. To obtain the optimal *NTM,* we can use *PM* and *PTM* as inputs to learn the model.

Previous studies like [23] indicate that the gradient descent (GD) algorithm can update the model parameters θ continuously to achieve the minimum value, along the opposite direction $-\nabla\theta J(\theta)$ of the gradient of the loss function $J(\theta)$. The step size can also be seen as a learning rate α.

The hypothesis function of the gradient descent algorithm can be set to:

$$h_\theta x = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n \tag{12}$$

The cost function is:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 \tag{13}$$

We should find a series of parameters θ that can minimize the cost function, where x_i is the sample feature matrix and y_i is the sample target value matrix. Each parameter θ_j can be updated as below:

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \tag{14}$$

The pseudo-code of this algorithm is described in Algorithm 1.

Algorithm 1 Gradient descent detection: GD_D(X, Y, α).

```
1: Input sample feature matrix X, sample target value matrix Y, learning rate α;
2: for v in Y
3:    v:=log(e)v
4: Repeat until convergence{
5:    for i=1 to m (Number of samples){
6:          θ_j := θ_j + α(y^(i) − h_θ(x^(i)))x_j^(i) (for every j to n (Number of features))
7:    }
8: }
9: for v in θ
10:   v:=exp(v)
11: return θ
```

The support vector machine (SVM) is a kind of supervised learning algorithm. It can help analyze data and identify patterns. In [24], the linear regression model is to adjust each sample data (x_i,y_i) in the training set according to a linear model $f(x)=\theta^T x+b$. The optimization objective function of SVM is $\min\frac{1}{2}\|\theta\|_2^2$. In this model, the loss should be zero when $f(x)$ is the same as y. SVM can define a constant ε as the deviation threshold between $f(x)$ and y. To summarize, the loss function for the SVM model is:

$$\mathrm{err}(x_i,y_i)=\begin{cases}0, & \left|y_i-\theta^T x_i-b\right|\le\ \varepsilon\\ \left|y_i-\theta^T x_i-b\right|-\varepsilon, & \left|y_i-\theta^T x_i-b\right|>\ \varepsilon\end{cases} \tag{15}$$

where x_i is the sample feature matrix, y_i is the sample target value matrix. We need to calculate a series of parameters θ that can minimize the loss. This can be treated as a problem of finding the minimum value under the constraint condition. The Lagrange multiplier is introduced to solve this problem and the regression equation is shown as below.

$$f(x)=\sum_{i=1}^{m}\left(\alpha_i-\alpha_i^*\right)K(x_i,x)+b \tag{16}$$

where α_i and α_i^* are the Lagrange multipliers, K is the kernel function. The pseudo-code of SVM algorithm is described in Algorithm 2.

Algorithm 2 Support vector machine detection: SVM_D(X, Y, α, K).

```
1: Input sample feature matrix X, sample target value matrix Y, learning rate α, specific kernel function 2: K;
3: for v in Y
4:    v:=log(e)v
5: Repeat until convergence {
6:    a. Using the heuristic method to do the optimization of the two variables α_i, α_i*, and use
7:    them to update the value of b;
8:    b. Using one sample per iteration to update θ;
9:    For i=1 to m{
10:         θ_j := θ_j + α(y^(i) − f(x^(i)))x_j^(i) (for every j)
11:   }
12: }
13: for v in θ
14:   v:=exp(v)
15: return θ
```

Fig. 3 shows an example to explain the process of our GD and SVM algorithm. The network contains six nodes and seven paths. We assume that N_4 is a malicious node and can launch an attack with a certain probability, in which it may modify the data packet or inject malicious data into the probe packets. We use N_i.T to represent the reputation value of the node. First, the probe packets are injected from S, and the probe packets are transmitted to D through multiple paths. By analyzing the path recorded in the probe packets and the probe packets' integrity, D can obtain the path information $\langle N_2, N_4\rangle$ and can update the trust value of this path Pa_i.T. We assume that a total of 200 probe packets are transmitted on the path Pa_i, in which 146 probe packets are complete and unmodified. We then have $Pa_i.\ T=\frac{146}{200}=N_2.T*N_4.T$, the equation can also be expressed as $\ln Pa_i.T=ln\frac{146}{200}=\ln N_2.T+\ln N_4.T=0*lnN_1.T+1*lnN_2.T+0*lnN_3.T+1*lnN_4.T$, in which 0 and 1 can indicate whether the node is included in this path. For the entire network, the sink extracts all paths' information from the probe packets transmitted via the paths.

$$PM=\ (Pa_1,Pa_2,Pa_3,\ldots,Pa_6,Pa_7)=\begin{pmatrix}1&1&0&0&0&0&0\\0&1&1&1&1&0&1\\0&0&0&0&1&1&1\\0&0&0&1&0&1&1\end{pmatrix}$$

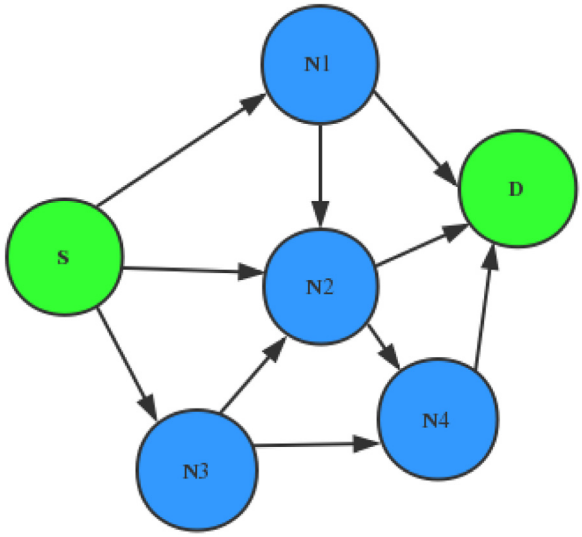

Fig. 3. An example network.

Then by analyzing the probe packets, the sink can obtain the trust values of all paths.

$$PTM = \left(ln\frac{200}{200}, ln\frac{200}{200}, ln\frac{200}{200}, ln\frac{146}{200}, ln\frac{200}{200}, ln\frac{174}{200}, ln\frac{134}{200}\right)$$

The reputation matrix of all nodes can be denoted by $NTM = (lnN_1.T, lnN_2.T, lnN_3.T, lnN_3.T)$, we have $NTM^*PM = PTM$. Both *PM* and *PTM* are used as inputs to the learning model. Then we get *NTM* through the GD learning model, and obtain the node's reputation, i.e., $(N_1.T, N_2.T, N_3.T, N_4.T) = (1.5878272192605, 1.0021171334548, 0.9961159711118, 0.7058899127776)$.

3.4. K-*means clustering*

After obtaining the reputation of all nodes, we can distinguish malicious nodes according to *N.T*. In the literature, *K*-means is a widely used clustering method. In this work, we use the *K*-means method to cluster nodes into different groups according to trust values. In our model, we only need to select $k = 2$ in *K*-means to directly cluster the nodes into two groups, namely benign group (BG) and malicious group (MG). However, the reputation of a node may be affected by the behavior of other nodes in its associated multi-hop path. For example, a malicious node could cause the trust values of some normal nodes in the same path to decrease at an intermediate level and may be assigned to the malicious group, resulting in a false rate. Therefore, in this work, we cluster all nodes into three groups, such as benign group (BG), unknown group (UG) and malicious group (MG). The intermediate level of *N.T* can be assigned to the unknown group. To further reduce the impact of other nodes in the same routing path on the reputation of nodes in UG, we further design an enhanced detection scheme to calculate the trust values of all nodes in a more accurate way.

3.5. *Enhanced detection*

To avoid the influence of nodes in the same routing path, we separate nodes in UG from other nodes and assign the nodes in UG to the discrete paths as possible. For example, assume N_1, N_2 are two nodes with medium reputation and there are some relevant paths: $Pa_1 = \langle S, N_1, D\rangle$, $Pa_2 = \langle S, N_2, D\rangle$ and $Pa_3\langle S, N_1, N_2, D\rangle$. In this case, we select Pa_1 and Pa_2 instead of Pa_3 because N_1 and N_2 are in the same path in Pa_3. It is known that the fewer nodes in the UG under the same path, the smaller the influence of other nodes could be. Then we re-inject new packets into these discrete paths, and collect new information from the sink. Later, we can build a new set of equations to train the model again and obtain the reputation of all nodes. Then we use the *K*-means method to cluster all nodes into two groups, namely, the final benign group (FBG) and the final malicious group (FMG).

In order to enhance detection and generate enhanced routing paths, for each node in the UG, we first should find all the paths containing this node in the original path set, here referred to as UP, then search the path in UP containing the fewest nodes in the MG and the fewest nodes in the UG, while adding this path to the enhanced path set. As shown in Fig. 4, after calculating the values of all nodes and we can have $UG = \{N_4, N_{10}, N_{13}\}$ and $MG = \{N_7, N_9\}$ based on the *K*-means clustering. For each node in UG, we can choose $Pa_1 = \langle S, N_2, N_4, N_5, N_8, N_{12}, N_{14}, N_{15}, D\rangle$, $Pa_2 = \langle S, N_2, N_4, N_6, N_{10}, N_{12}, N_{14}, N_{15}, D\rangle$ and $Pa_3 = \langle S, N_1, N_3, N_5, N_8, N_{11}, N_{13}, N_{15}, D\rangle$ to join the enhanced path set.

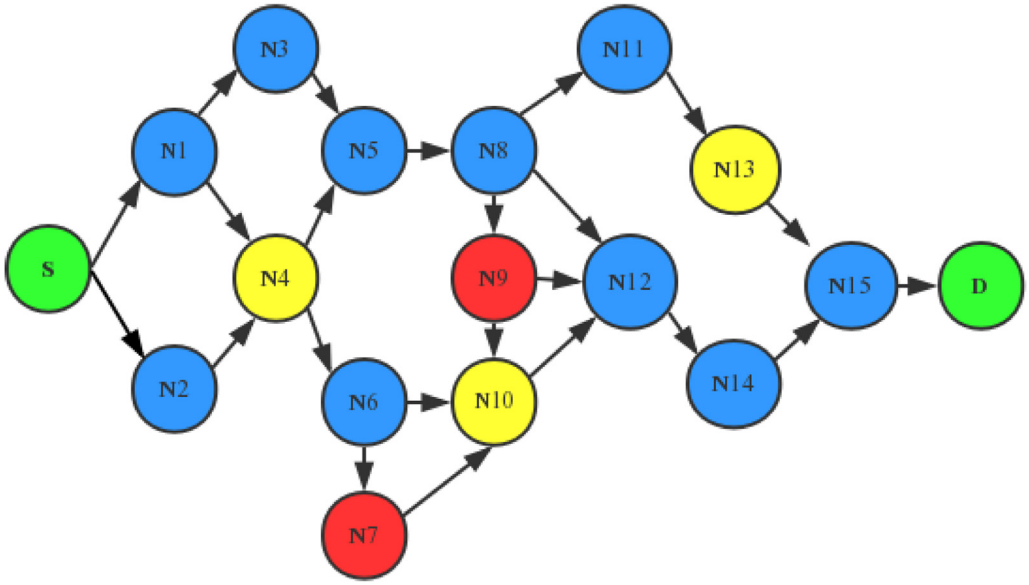

Fig. 4. Network nodes' distribution and nodes' information, BG nodes are in blue, UG nodes are in yellow and MG nodes are in red. *S, D* are trusted source and sink in green. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. Experimental results

In this section, we compare the Hard Detection (HD) [21] with our proposed GD algorithm and SVM algorithm in terms of detection performance. For the HD algorithm, the reputation of all nodes in the same path are directly solved by the reputation of paths. Let k_i denote the number of paths containing node I, $k_{i,j}$ denote the jth path in the k_i path, $|k_{i,j}|$ denote the node's number of $k_{i,j}$, and $Pa.T_{i,j}$ denote the reputation of $k_{i,j}$. Then the node's reputation in the HD can be calculated as:

$$N_i.\mathrm{T} = \frac{1}{k_i}\sum_{j=1}^{k_i} \sqrt[|k_{i,j}|]{Pa.T_{i,j}} \tag{17}$$

In order to evaluate our enhanced detection scheme, we exploit our algorithms in two different conditions: with and without the enhanced detection scheme. To summarize, we compare the detection performance among HD, GD, the enhanced GD (GD_ED), SVM and the enhanced SVM (SVM_ED).

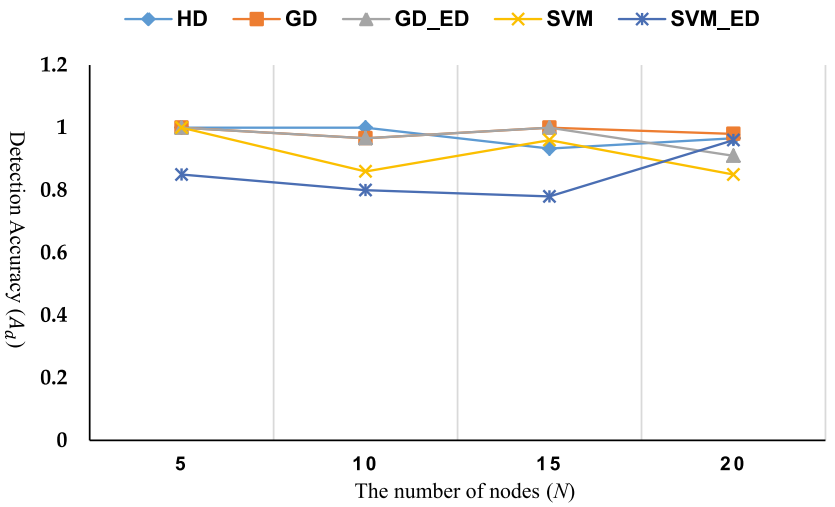

Fig. 5. The impact of the number of nodes on accuracy A_d: the number of nodes is set to 5, 10, 15 and 20 respectively in this experiment, and it is found that GD, GD_ED and HD have higher detection accuracy than others.

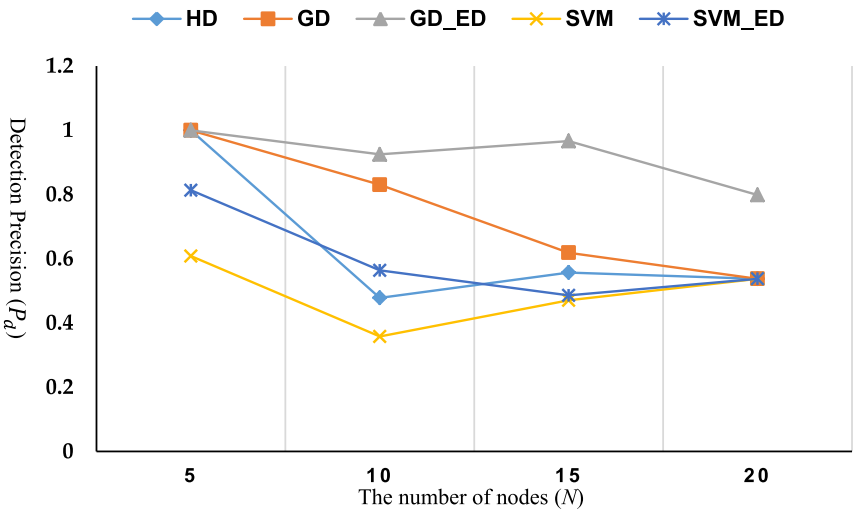

Fig. 6. The impact of the number of nodes on detection precision P_d: the number of nodes is set to 5, 10, 15 and 20 respectively in this experiment, and GD_ED have higher detection precision than others.

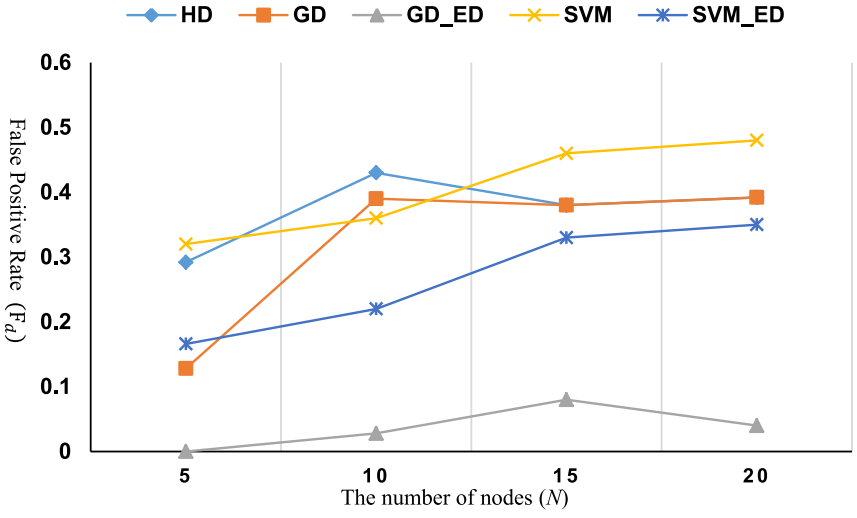

Fig. 7. The impact of the number of nodes on false positive rate F_d: the number of nodes is set to 5, 10, 15 and 20 respectively in this experiment, and it is found that GD_ED and SVM_ED have a lower false positive rate.

In the simulation, we assume that there are N relay nodes uniformly distributed in a 100×100 m^2 rectangle. The communication range of each node is $r = 10$ m. In our experiment, we generate 10 random networks and the experiment was run in 10 rounds. We evaluate the performance in the aspects of detection accuracy, detection precision and false positive rate. The detection accuracy is defined as A_d indicating the number of malicious nodes correctly identified divided by the number of malicious nodes. The detection precision is defined as P_d indicating the number of malicious nodes correctly identified divided by the number of all nodes. The false positive rate is defined as F_d indicating the number of benign nodes identified as malicious nodes divided by the number of benign nodes. These metrics are summarized as follows:

4.1. The impact of the number of nodes

To evaluate the impact of the number of nodes on the detection accuracy, detection precision and the false positive rate, we set the number of nodes to 5, 10, 15 and 20, respectively. The number of our injected packets is 10,000; the probability of attack is 0.3; the proportion of malicious nodes is 0.3 and the diversity of the network is set to use all reachable paths. The relationship between $A_d(P_d, F_d)$ and the number of nodes is plotted in Figs. 5–7.

On the whole, it is found that the detection accuracy of HD, GD, GD_ED and SVM can reach 90% while SVM_ED is relatively low. When the number of nodes increases, the detection accuracy can be achieved more accurately, i.e., higher than 90%. Meanwhile GD_ED can reach a low false detection rate, i.e., less than 5%.

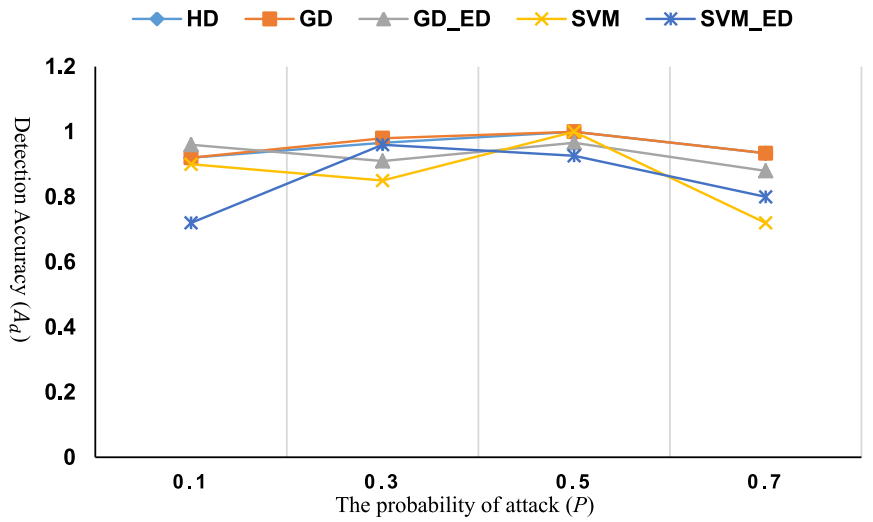

Fig. 8. The impact of the probability of attack on accuracy A_d: the probability of attack is set to 0.1, 0.3, 0.5 and 0.7 respectively, and the detection accuracy of HD and GD can be higher than other detection methods.

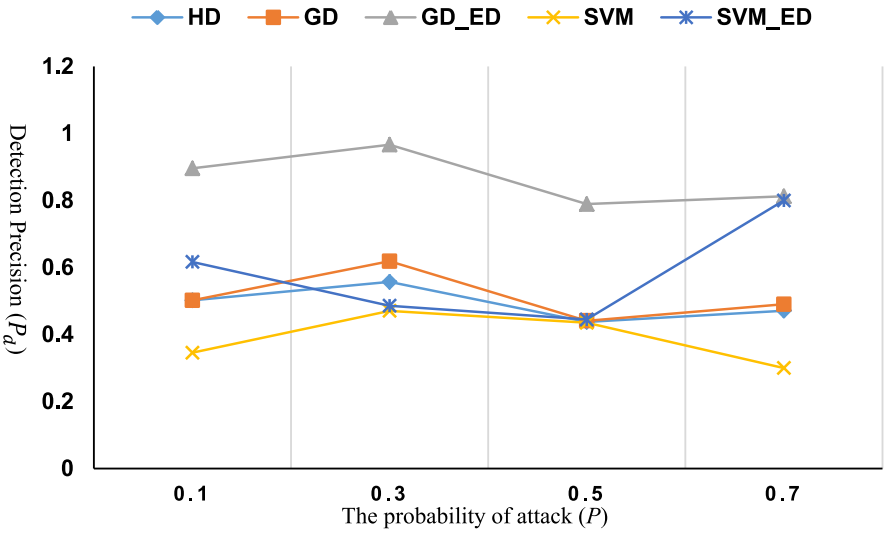

Fig. 9. The impact of the probability of attack on detection precision P_d: the probability of attack is set to 0.1, 0.3, 0.5 and 0.7 respectively, and it is found that the detection precision of GD_ED is higher than the other detections method. GD_ED could keep the rate over 80%.

4.2. The impact of the probability of attack

In this experiment, we evaluate the impact of node's attack probability on the detection accuracy, detection precision and the false positive rate. We set the node's attack probability to be 0.1, 0.3, 0.5 and 0.7 respectively. The node number is set as 15; the number of injected packets is 10,000; the proportion of malicious nodes is 0.3; the diversity of the network is set to use all reachable paths. The relationship between $A_d(P_d, F_d)$ and the node attack probability is plotted in Figs. 8–10.

It is found that when the probability is either too low or too high, it is very difficult to reach good detection accuracy. Given the appropriate attack probability, all methods can reach a detection accuracy rate over 90%; and GD_ED could achieve a higher rate than others. For example, GD_ED could reach an average error rate of less than 5%.

4.3. The impact of the proportion of malicious nodes

In this experiment, we evaluate the impact of the number of malicious nodes on the detection accuracy, detection precision and the false positive rate. We set the proportion of malicious nodes in all nodes to be 0.1, 0.3, 0.5 and 0.7 respectively. The number of nodes is set as 15; the number of injected packets is set as 10,000; the probability of attacks is 0.3 and the

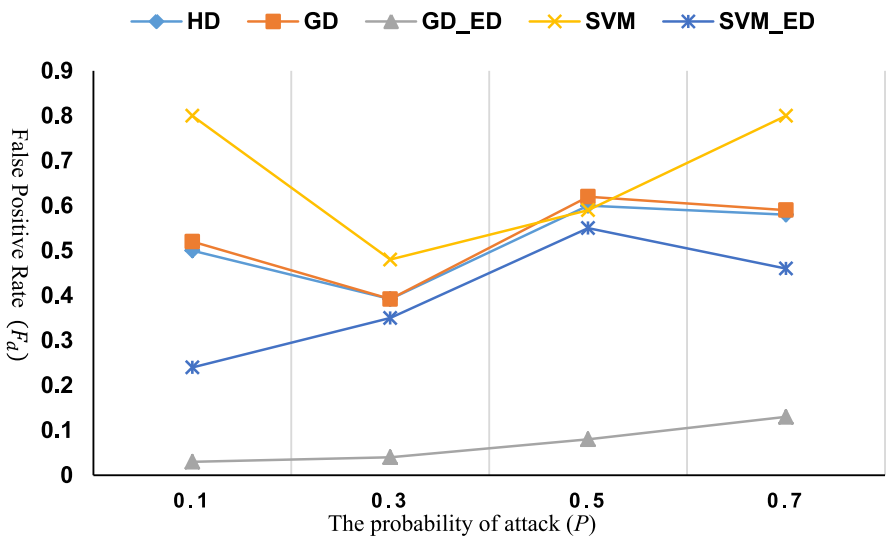

Fig. 10. The impact of the probability of attack on false positive rate F_d: the probability of attack is set to 0.1, 0.3, 0.5 and 0.7 respectively, and it is found that false positive rate of GD_ED could reach an average rate of less than 5%.

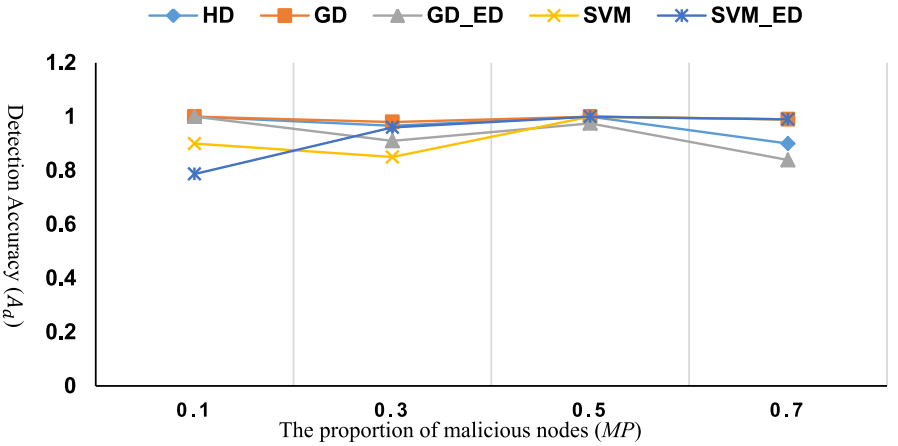

Fig. 11. The impact of the proportion of malicious nodes on accuracy A_d: it is found that the detection accuracy rates of all detection methods are similar, while the detection accuracy of GD is stable and higher than other detection methods.

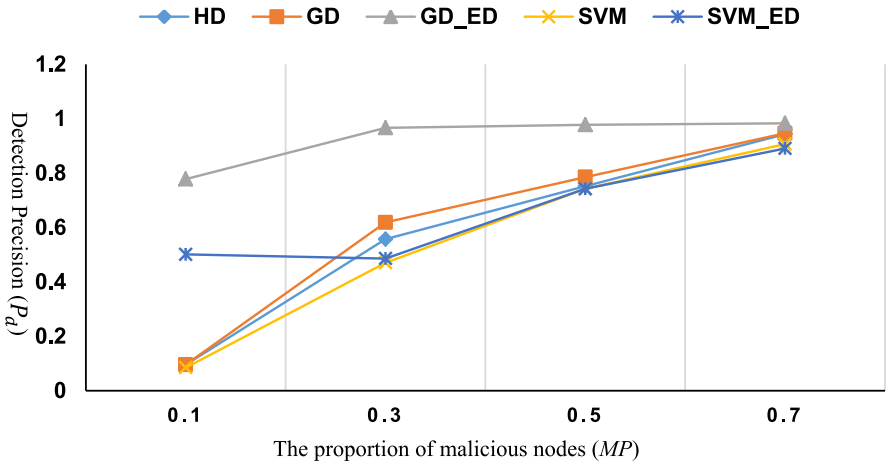

Fig. 12. The impact of the proportion of malicious nodes on detection precision P_d: it is found that the detection precision of GD_ED is stable and higher than other detection methods.

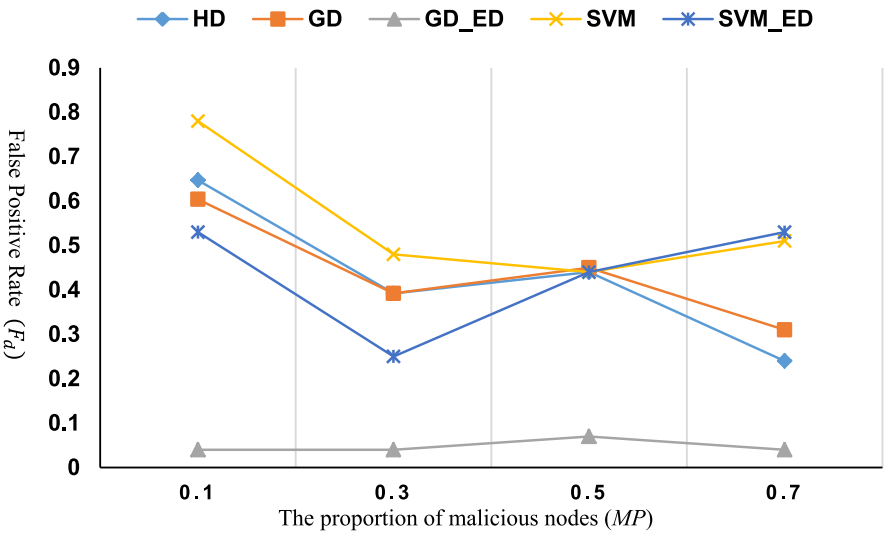

Fig. 13. The impact of the proportion of malicious nodes on false positive rate F_d: it is observed that the more malicious nodes, the lower the false detection rate. The overall performance of GD_ED is better than other methods.

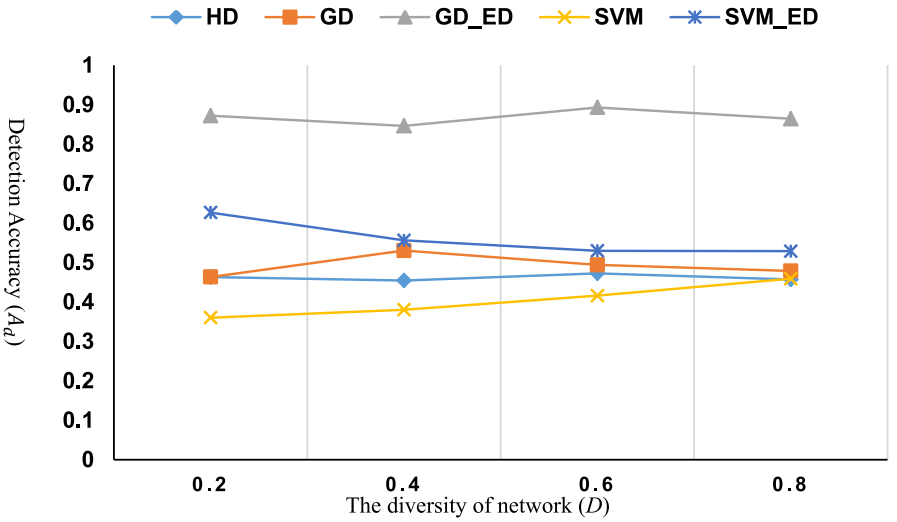

Fig. 14. The impact of the diversity of network on accuracy A_d: it is found that the detection accuracy of GD is stable and higher than other detection methods.

diversity of the network is set to use all reachable paths. The results regarding $A_d(P_d, F_d)$ and the number of malicious nodes are plotted in Figs. 11–13.

Figs. 11 and 12 show that with appropriate number of malicious nodes, the detection accuracy of all methods can reach more than 90% in the end. While GD_ED can achieve stable and higher detection accuracy than other methods, i.e., its false detection rate could reach less than 5%.

4.4. The impact of the diversity of network

In this experiment, we evaluate the impact of the diversity of network on our evaluation metrics. We set the proportion of reachable paths in all paths to be 0.2, 0.4, 0.6 and 0.8, respectively. The number of nodes is set as 15; the number of injected packets is set as 10,000; the proportion of malicious nodes is 0.3 and the probability of attack is 0.3. The relationship between $A_d(P_d, F_d)$ and path diversity is plotted in Figs. 14–16.

It is found that with appropriate number of paths, the detection accuracy of GD_ED could reach a rate of more than 90% with a better error rate than other methods, i.e., its error rate can achieve a rate of less than 5%.

Overall, our simulation results demonstrate that our approach can achieve an overall detection rate of 90% and above in a common scenario. While under the enhanced detection scheme, our methods can achieve both higher detection accuracy

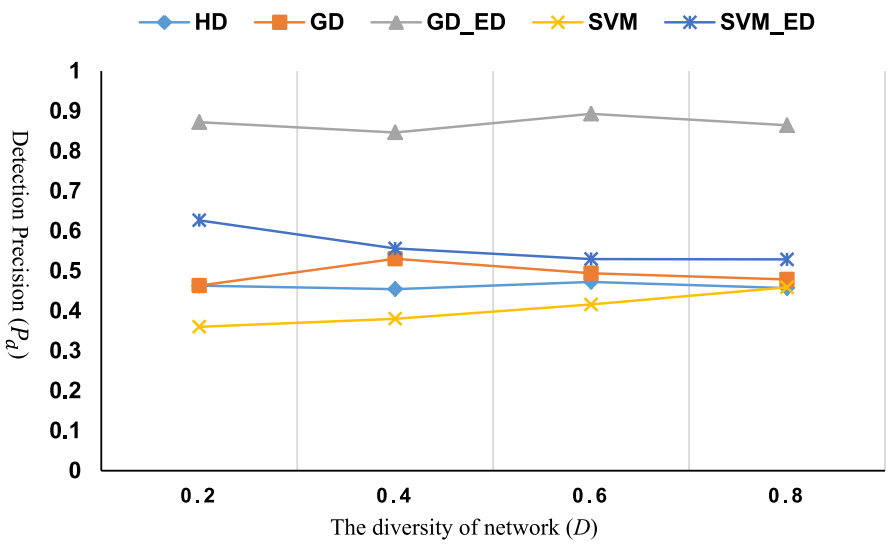

Fig. 15. The impact of the diversity of network on detection precision P_d: it is found that the detection precision of GD_ED is stable and higher than the other detection methods.

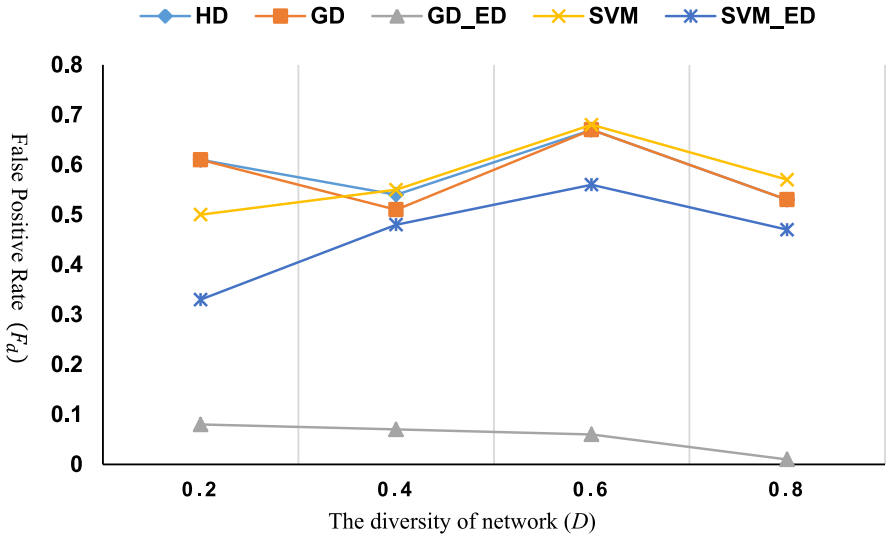

Fig. 16. The impact of the diversity of network on false positive rate F_d: it is observed that GD_ED could reach the best performance, i.e., its error rate is less than 5%.

and lower false detection rate, i.e., the detection precision of GD_ED method is stable and higher than other methods, with a false detection rate of 5% or less. In this case, we consider that our detection methods can help identify malicious nodes efficiently and accurately. In our future work, we plan to consider more related studies in this area like [25] and perform a larger comparison.

5. Concluding remarks

IoT network is developing at a fast pace, but it is also threatened by various attacks. For example, insider attacks are one of the big threats in an IoT network, in which cyber intruders can control an internal node to access the system and network resources. There is a need to detect and identify internal malicious nodes in an effective way. In this paper, we focus on insider attacks and aim to detect malicious nodes based on the reputation of both routing paths and nodes. We propose two algorithms of gradient descent (GD) and support vector machine (SVM) to learn the reputation value of insider nodes and then cluster these nodes into benign group and malicious group using the *K*-means method. In order to improve the detection accuracy, we further enhance the process of algorithm learning by optimizing the routing path and compute the ultimate trust values for all nodes in an IoT environment. The experimental results show that our method can detect

malicious nodes with more than 90% accuracy, and the enhanced detection scheme could reach an even lower false detection rate, i.e., less than 5%. For our current work, it requires reliable source nodes to inject probe packets. The data collection and analysis also require some time to complete, which may not be able to perform the real-time detection. In our future work, we plan to use online learning algorithm to learn the reputation of nodes in real time.

Conflict of interest

None.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 61402225; the China Postdoctoral Science Foundation under Grant No. 2013M540447; the Jiangsu Postdoctoral Science Foundation under Grant No. 1301020C. This work was also partially supported by the Foundation of State Key Laboratory for smart grid protection and operation control; the Science and Technology Funds from National State Grid Ltd. (The Research on Key Technologies of Distributed Parallel Database Storage and Processing based on Big Data).

References

[1] Javed F, Afzal MK, Sharif M, Kim B-S. Internet of things (IoT) operating systems support, networking technologies, applications, and challenges: a comparative review. IEEE Commun Surv Tuts 2018;20:2062–100. https://doi.org/10.1109/COMST.2018.2817685.
[2] Corak BH, Okay FY, Guzel M, Murt S, Ozdemir S. Comparative analysis of IoT communication protocols. In: 2018 International symposium on networks, computers and communications (ISNCC); 2018. p. 1–6. https://doi.org/10.1109/isncc.2018.8530963.
[3] Tseng F-H, Chiang H-P, Chao H-C. Black hole along with other attacks in MANETs: a survey. J Inf Process Syst 2018;14:56–78. https://doi.org/10.3745/JIPS.03.0090.
[4] Jan MA, Nanda P, Liu RP. A sybil attack detection scheme for a forest wildfire monitoring application. Fut Gener Comp Syst 2018;80:613–26. https://doi.org/10.1016/j.future.2016.05.034.
[5] Abdelhakim M, Lightfoot L, Ren J, Li T. Reliable communications over multi-hop networks under routing attacks. GLOBECOM 2015:1–6. https://doi.org/10.1109/GLOCOM.2015.7417404.
[6] Abdelhakim M, Liu X, Krishnamurthy P. Diversity for detecting routing attacks in multi-hop networks. In: 2018 International conference on computing, networking and communications (ICNC); 2018. p. 712–17. https://doi.org/10.1109/ICCNC.2018.8390382.
[7] Rana KG, Cai Y, Azeem M, Ditta A, Yu H, Khuhro SA. Wireless ad hoc network: detection of malicious node by using neighbour-based authentication approach. Int J Wirel Mob Comput 2018;14(1):16–24. https://doi.org/10.1504/ijwmc.2018.10011093.
[8] Rikli N-E, Alnasser A. Lightweight trust model for the detection of concealed malicious nodes in sparse wireless ad hoc networks. Int J Distrib Sens Netw 2016;12(7). https://doi.org/10.1177/1550147716657246.
[9] Chen Z, Tian L, Lin C. Trust model of wireless sensor networks and its application in data fusion. Sensors 2017;17(4):703. https://doi.org/10.3390/s17040703.
[10] Romman AA, Al-Bahadili H. Performance analysis of the neighbor weight trust determination algorithm in MANETs. Int J Netw Secur Appl 2016;8(4):29-40. https://doi.org/10.5121/ijnsa.2016.8403.
[11] Ahmed S, Tepe K. Recommendation trust for improved malicious node detection in ad hoc networks. In: IEEE 86th VTC-Fall; 2017. p. 1–5. https://doi.org/10.1109/VTCFall.2017.8288217.
[12] Zhang W, Zhu S, Tang J, Xiong N. A novel trust management scheme based on dempster-shafer evidence theory for malicious nodes detection in wireless sensor networks. J Supercomput 2018;74(4):1779–801. https://doi.org/10.1007/s11227-017-2150-3.
[13] Liu X, Abdelhakim M, Krishnamurthy P, Tipper D. Identifying malicious nodes in multihop IoT networks using dual link technologies and unsupervised learning. Open J Internet Things 2018;4(1):109–25. https://www.ronpub.com/ojiot/OJIOT_2018v4i1n09_XinLiu.html.
[14] Zawaideh F, Salamah M. An efficient weighted trust-based malicious node detection scheme for wireless sensor networks. Int J Commun Syst 2019;32(3). https://doi.org/10.1002/dac.3878.
[15] Kaplantzis S, Shilton A, Mani N, Sekercioglu YA. Detecting selective forwarding attacks in wireless sensor networks using support vector machines. In: IEEE ISSNIP; 2008. p. 335–40. https://doi.org/10.1109/ISSNIP.2007.4496866.
[16] Akbani R, Korkmaz T, Raju GVS. A machine learning based reputation system for defending against malicious node behavior. GLOBECOM 2008:2119–23. https://doi.org/10.1109/GLOCOM.2008.ECP.408.
[17] Nahiyan K, Kaiser S, Ferens K, McLeod R. A multi-agent based cognitive approach to unsupervised feature extraction and classification for network intrusion detection. In: ACC'17; 2017. p. 25–30. https://csce.ucmss.com/cr/books/2017/LFS/CSREA2017/ACC6041.pdf.
[18] Dromard J, Roudiere G, Owezarski P. Online and scalable unsupervised network anomaly detection method. IEEE Trans Netw Serv Manag 2017;14(1):34–47. https://doi.org/10.1109/TNSM.2016.2627340.
[19] Luo T, Nagarajan SG. Distributed anomaly detection using autoencoder neural networks in WSN for iot. In: 2018 IEEE International Conference on Communications (ICC); 2018. p. 1–6. https://doi.org/10.1109/ICC.2018.8422402.
[20] Ayadi A, Ghorbel O, BenSaleh MS, Obeid A, Abid M. Outlier detection based on data reduction in WSNs for water pipeline. SoftCOM 2017:1–6. https://doi.org/10.23919/SOFTCOM.2017.8115570.
[21] Liu X, Abdelhakim M, Krishnamurthy P, Tipper D. Identifying malicious nodes in Multi-hop IoT networks using diversity and unsupervised learning. In: IEEE International Conference on Communications; 2018. p. 1–6. https://doi.org/10.1109/icc.2018.8422484.
[22] Wang C, Hussain SR, Bertino E. Dictionary based secure provenance compression for wireless sensor networks. IEEE Trans Parallel Distrib Syst 2016;27(2):405–18. https://doi.org/10.1109/TPDS.2015.2402156.
[23] Haykin S. Neural networks and learning machines, vol. 3, Upper Saddle River NJUS: Pearson Education; 2009. 07458 https://cours.etsmtl.ca/sys843/REFS/Books/ebook_Haykin09.pdf.
[24] Schölkopf B, Smola AJ. Learning with Kernels: support vector machines, regularization, optimization, and beyond. Cambridge MAUS: MIT Press; 2002. p. 1–626. https://doi.org/10.1109/TNN.2005.848998.
[25] Li W, Meng W, Kwok LF. Investigating the influence of special on-off attacks on challenge-based collaborative intrusion detection networks. Fut Internet 2018;10(1):1–16.

Liang Liu is currently a lecture in College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu Province, China. His research interests include distributed computing, big data and system security. He received the Ph.D. degree in computer science from Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu Province, China in 2012.

Jinxiu Yang is currently a master student in College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu Province, China. She received her bachelor's degree in Nanjing Normal University in 2019. Her research focuses on Internet of things (IoT) security.

Weizhi Meng is currently an assistant professor in the Cyber Security Section, Technical University of Denmark (DTU), Denmark. His primary research interests are cyber security and intelligent technology in security, including intrusion detection, smartphone security, biometric authentication, HCI security, trust computing, blockchain in security, and malware analysis. He served as program committee members for 50+ international conferences.