Yan *et al. J Wireless Com Network* (2023) 2023:25
https://doi.org/10.1186/s13638-023-02233-8

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

A novel Sybil attack detection scheme in mobile IoT based on collaborate edge computing

Junwei Yan[1], Tao Jiang[2], Liwei Lin[3,4*], Zhengyu Wu[5], Xiucai Ye[6], Mengke Tian[7,8] and Yong Wang[8]

*Correspondence:
llw02@fjut.edu.cn

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
[2] Henan University, Henan, China
[3] School of Computer Science and Mathematics, Fujian University of Technology, Fujian 350118, China
[4] Fujian Key Laboratory of Big Data Mining and Applications, Fujian 350118, China
[5] Honor Device Co., Ltd., Shenzhen, China
[6] Department of Computer Science, University of Tsukuba, Tsukuba, Ibaraki, Japan
[7] School of Integrated Circuits, Peking University, Beijing 100871, China
[8] Beijing Microelectronics Technology Institute, Beijing 100076, China

Abstract

Background: Internet of things (IoT) has been used in many places. IoT make devices connected to the Internet via sensor devices to achieve the interconnection between things and things, people and things. Sybil attacker attacks IoT by imitating the identity of users. Few methods are applicable for mobile IoT in previous Sybil attack detecting methods, while the methods are mainly focus on static IoT.

Results: A distributive and lightweight Sybil attack detection scheme in the mobile IoT is proposed in this paper. This scheme works around received signal strength indications (RSSI). The scheme consists of two rounds. Identity information is sent from member nodes to edge nodes in both of the two rounds. In the first round edge, nodes calculate the possible RSSI interval for each member node; in the second round, they check the RSSI value of member nodes to detect Sybil attacks. Intelligent algorithms are used to predict the position of member nodes, which makes the theoretical interval more accurate. Extensive experimental studies show that in the true and false detection rate, this scheme is superior to many existing schemes.

Keywords: Sybil attack, Mobile IoT, Edge Computing

1 Introduction

IoT is a connection of many technologies, such as communication, sensory, information processing, and networking, that make huge amounts of devices connected [1]. IoT has been placed in many places, such as industrial automation, emergency rescue, transportation, and health care [2]. The large scale of IoT requires distributed management, and edge compute is an effective way to make the management of IoT distributed [3, 4]. IoT consisted of static sensor nodes in the early stage [5]. However, with the progress of mobile communication technology and the growing demands for mobile social interaction, the application of mobile nodes is more and more extensive.

Security is an important issue that cannot be ignored in many systems, and IoT is no exception. In October 2016, a distributed denial-of-service (DDoS) attack by malware Mirai was launched on service provider Dyn using an IoT botnet. That brought down large parts of the Internet, including Twitter, Netflix, the Guardian, Reddit and CNN.

The Sybil attack is a well-known and destructive cyberattack in IoT affecting the network layer [6]. Sybil attack legitimizes malicious nodes in a network by using multiple fake identities, and it can be used in conjunction with other attacks such as message suppression and channel jamming attacks. [7].

Some countermeasures to detect Sybil attacks have been proposed by researchers, for example, time difference of arrival (TDOA) [8], neighboring information [9], RSSI [10], random key pre-distribution and radio resource testing [11], angle of arrival (AOA) [12]. Most of these methods are weak for mobile IoT because they are based on neighbor cooperation or node position. There are a few proposals in mobile IoT: Piro et al. [13] proposed 2 algorithm based on observer monitoring named PASID and PASID-GD. These methods may slow down the performance of sensor nodes because they occupy much memory overhead. Jamshidi et al. [14] proposed a lightweight method based on nodes mobile behavior. This method relies on historical records so that it has poor stability and robustness.

We propose a detection scheme for detect Sybil attacks in mobile IoT based on edge computing to overcome those shortcomings. Sybil attack detection may be difficult for normal member nodes because it requires storing and analyzing a large amount of feature data. Edge nodes have large storage space and strong computing power, which are suitable for our problem. Due to the short distance between member nodes and edge nodes, member nodes can be directly managed to reduce communication delay. In the cloud-based IoT system, the data of nodes need to be transmitted in the data center for long distance [15], which has low scalability. In [16, 17], two algorithms are proposed to optimize data communication in cloud computing. Using edge nodes to apply distributed management improves scalability compared to cloud-based IoT systems. In our approach, when each detection is completed, sensor nodes will clear most of the data to reduce the dependence on historical data. The RSSI value is selected as the identifier to distinguish malicious member nodes from normal member nodes for the accuracy issue. Specifically, in mobile IoT, when member nodes move, the RSSI values change continuously. Therefore, in this paper, it is a crucial part to determine the theoretical range of RSSI.

The follows are the contributions of this paper:

- This paper designed a detection scheme for Sybil attacks in the mobile IoT. Distinguishing identifiers of member nodes are formed through the member nodes information including RSSI, which can detect malicious nodes by further analyzing this identifier. By studying the fluctuation feature of RSSI values over a period of time, the RSSI's theoretical range is given. The intelligent algorithm is used to predict the positions of member nodes, which makes the theoretical interval of RSSI more accurate. Experimental results show that the detection accuracy of the proposed detection scheme is well.
- To reduce power, computation and memory overhead, edge computing is used. In the detection process, the edge node with better performance and no other tasks undertakes the computing task. The proposed method does not store the historical data of all nodes and in memory overhead our proposal is superior to other detection methods.

We start by reviewing the related work of detect Sybil attacks (Sect. 2) and discussing the basic models (Sect. 3). Then, we present the detailed detection scheme and evaluate the performance of the scheme (Sect. 4). We simulate the model (Sect. 5) and then conclude the paper (Sect. 6).

2 Related work

The shortcomings of most previous Sybil attack detection work are summarized as bellow.

2.1 Static IoT

Work [11] proposed several schemes to detect Sybil attacks, including code attestation (CA), identity registration (IR), position verification (PV), random key pre-distribution (RKP), and radio resource test (RRT). Work [8] uses TDOA ratio to detect malicious nodes. Work [18] proposes a malicious node identification scheme based on AOA, because malicious identities in one node share the physical location. The RSSI value of one physical node is the same, and some methods are based on RSSI. Work [19] estimates the location of nodes by using RSSI. Work [10] uses 4 sensors to locate the position of each node. The method proposed by [20] combines the method of the status of member nodes with RSSI. But methods above are not designed for mobile IoT. Work [21] proposed a method to against Sybil attack in routing protocol for low-power and lossy networks (RPL). Sybil attack is a kind of Byzantine attack, and there are other works against Byzantine attacks in the artificial intelligence field. Work [22] presented an alarming mechanism to build a Byzantine-robust federated learning system, and work [23] trains a Bayesian neural network via an adversarial distribution to improve the practical applications' performance. However, work [22] is aimed at federated learning, and work [23] is more about theoretical research, which is not IoT.

2.2 Mobile IoT

Work [13] proposed 2 algorithms based on observer monitoring named PASID and PASID-GD. In the two methods, they analyze packets that often appear together to flag suspicious nodes. Due to the large amount of data to be stored, the memory overhead of these two methods is small and may affect the normal functions of nodes. Work [14] and [24] detect mobile Sybil attacks via historical movement behaviors of nodes and therefore also may affect the normal functions over long time. Work [25] proposed a centralized Sybil attack detection scheme based on geographical location in mobile sensor networks, which included 3 stages: cluster nodes, select nodes near Sybil nodes and routing process. We can see this scheme is not proper. Work [26] proposed a method based on the registrations of base stations, so the scalability is not high enough. Work [27] proposed a detection scheme with watchdog nodes, which has the problems of large communication overhead. Work [28–30] proposed 3 schemes of vehicular ad hoc networks (VANET), but they are limited to VANET. Work [31] presented a memory-augmented autoencoder approach for detecting anomalies in IoT data, but it is not enough for Sybil attacks.

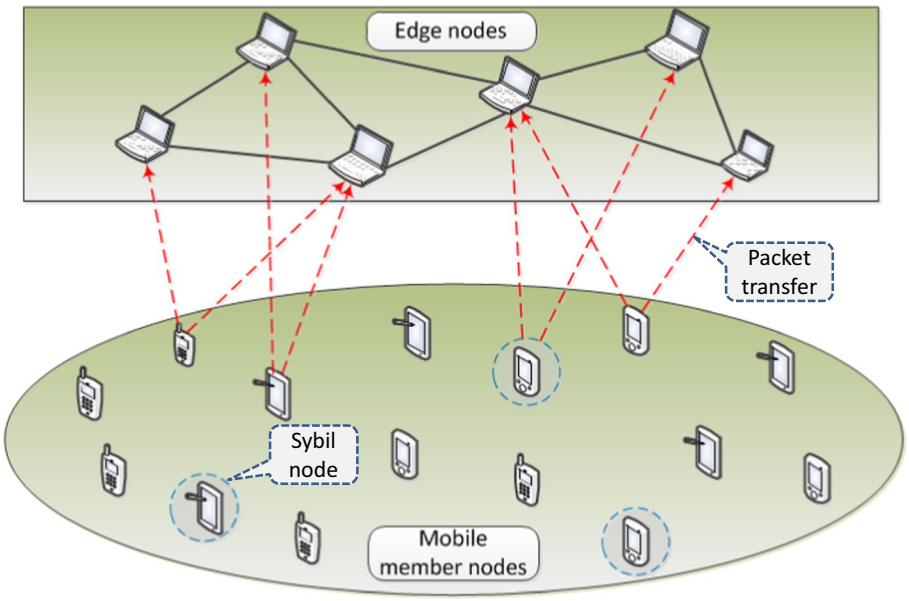

Fig. 1 Schematic diagram of Sybil attack

3 Background and system model

The problem, models and necessary notations of this paper are presented in this section.

3.1 Problem statement

N mobile nodes (u_k), including normal and malicious ones, form a single-hop wireless network as shown in Fig. 1.[1] M edge nodes (e_k) are high energy. All edge nodes have weak mobility. The transmission power of each node can be adjusted to reach a further node [32]. Assuming that by using the algorithm in [33] or global positioning system (GPS) the position of each node can be computed, we could use algorithms like [34] for privacy protection.

The adversary reprograms some normal nodes into malicious nodes that can forge multiple Sybil nodes. These nodes can influence other nodes in the network that are not limited to neighbor nodes. This paper uses two evaluation metrics: One is the true-positive rate (TPR): the rate that normal users classified as normal ones, and the other is the false positive rate (FPR): the rate that Sybil users classified as normal ones. We need to make a trade-off between these two rates. Computation and memory overhead of edge nodes are concerned due to the limited computing power and energy.

3.2 Network space channel model

Jakes model is used as the cyberspace channel model in this paper, which is mainly used in wireless communications [35, 36]. RSSI between e_i and u_j in the Jakes model is defined as

[1] No forwarding is required for communication between member nodes and edge nodes in this paper.

$$R_i^j = \frac{k * P_x}{(d_i^j)^\alpha}. \tag{1}$$

In this equation, k is a constant, P_x is the transmitted power, and d_i^j is the Euclidean distance between e_i and u_j, which is determined by

$$d_i^j = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}. \tag{2}$$

α is the range-power drop ramp, which may be different in different environments. α can be chosen according to the following rules: Free spaces are $\alpha = 2$; buildings with line-of-sight connections are $\alpha \in [1.6, 1.8]$; and the urban areas are $\alpha \in [2.7, 3.5]$ [37]. The signal strength is a function of the distance between two nodes which communicate with each other in Jakes channel space. Thus, the method of this paper is based on RSSI.

3.3 Mobility model

We assume that edge nodes remain stationary for fairly short time intervals due to the weak mobility of edge nodes with respect to member nodes, and the waypoint model is used to verify the algorithm's correctness in this paper.

4 Detection scheme and performance evaluation

This section provides the detailed detection scheme and provides the performance evaluation from three aspects: computation, memory and communication.

4.1 Details of the detection scheme

The detection finishes in 2 stages. Edge nodes collect member nodes' information in the 1st and 2nd round, and after that classify the member nodes in the judgment stage. Our detection scheme begins at time t_0. Edge nodes launch requirement to each member node u_i so that it has to send control packets containing its own identity to two of edge nodes which is nearest to it, e_1 and e_2. e_1 and e_2 use Eq. (1) to calculate R_1^{i1} and ${R_2}^{i1}$, respectively. Then, e_1 send packets including the value of ${R_1}^{i1}$ to e_2. Define the ratio of ${R_2}^{ir}$ to ${R_1}^{ir}$ in round x as

$$\eta_x = \frac{R_2^{ir}}{R_1^{ir}} = \left(\frac{d_1^i}{d_2^i}\right)^\alpha. \tag{3}$$

Then, e_2 calculates the ratio η_1 by Eq. (3). After that, e_2 calculates the theoretical interval of feasible η_2 value, denoted as I. We define the set of claimed identities of member nodes as ID_0.

At time t_1, the system launches order to each member node again. According to Eq. (3), e_2 calculates η_2. We delete the set of claimed identities of member nodes as ID_1.

After the above two rounds, e_2 identifies the member nodes through the identity descriptions collected in the first two rounds. There are three of the cases:

- $ID_0 = \{a, b, c, d\}$ $ID_1 = \{a, b, c, d\}$ The ids a, b, c, d in ID_0 are all normal nodes whose ids are the same in both ID_0 and ID_1.

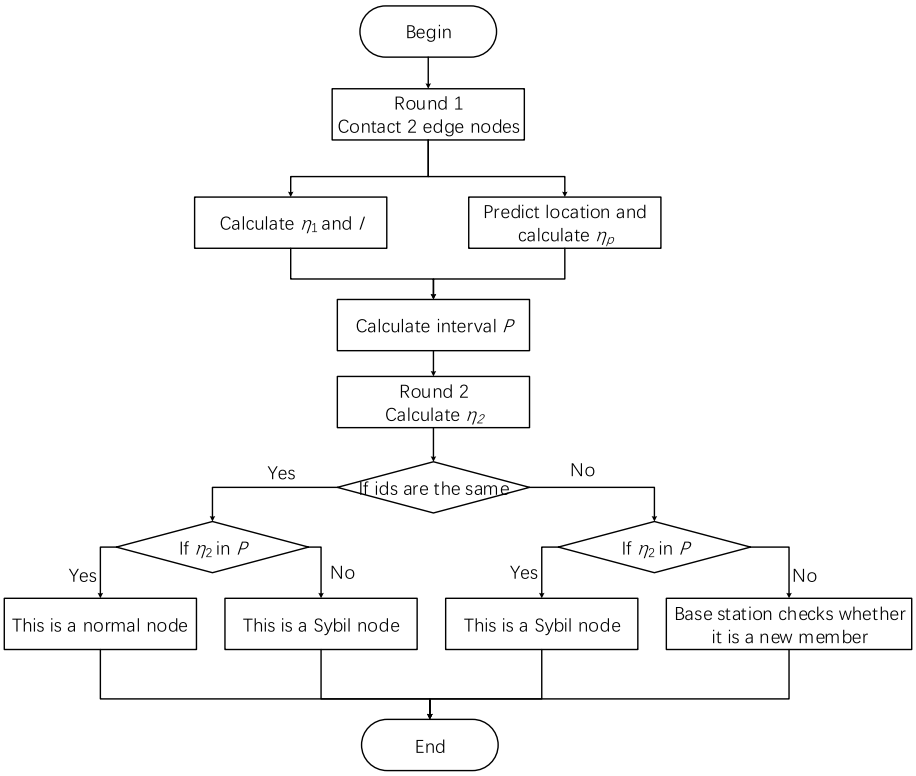

Fig. 2 Flowchart of the proposed algorithm

- $ID_0 = \{a, b, c, d\}$ $ID_1 = \{a, b, c, e\}$ The ids a, b, c in ID_0 are normal nodes whose ids are the same in both ID_0 and ID_1, while the id d in ID_0 is a Sybil node whose id changed to e.
- $ID_0 = \{a, b, c, d\}$ $ID_1 = \{a, b, c, e\}$ The ids a, b in ID_0 are normal nodes whose ids are the same in both ID_0 and ID_1, the id c in ID_0 is a Sybil node whose id changed to e, and the id c in ID_0 is a Sybil node whose id changed d.

Since there are far more normal nodes than malicious nodes, there should exist more pairwise ids than different ids, which in our model are more likely to be Sybil. Our flowchart for making judgments is shown in Fig. 2. If a Sybil node declares itself to be different id in the above two rounds, η_2 computed using Eq. (3) is theoretically considered to belong to the interval since it is the same node. In that case, edge node should judge it to be a Sybil node. Otherwise, the node can be a new member, so base stations are needed to check it. Another case where a node is classified as Sybil is when it moves too fast and causes case η_2 to go out of range of interval I.

4.2 The interval *I*

For convenience, we take two edge nodes e_1 and e_2, and one member node u_i for mathematical demonstration. Figure 3 abstracts our scheme as a mathematical model, which has a planar rectangular coordinate system with the center of e_1 and e_2 (C_1 and C_2 in the figure) as the origin and the line between the two edge nodes as the X-axis, and u_i in the time t_0 is shown as the point N in the figure.

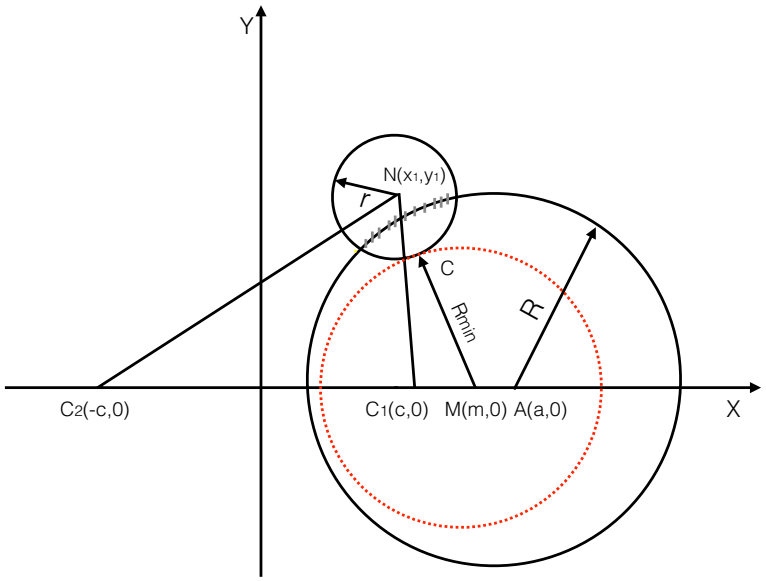

Fig. 3 Mathematical model for explaining the detection process

At time t_1, the circle O_1 is a possible position of u_i in the time t_1, whose center is N and radius is r; the locus of u_i where η_2 is fixed is an Apollonian circle with an radius of r centered on point A, since points C_1 and C_2 are fixed between t_0 and t_1. We delete it as O_2. We have

$$r = v_i \Delta t. \tag{4}$$

According to Eq. (3), η_2 is fixed when $\frac{NC_2}{NC_1}$ is fixed. We need calculate $\frac{NC_2}{NC_1}$ to calculate the interval I according to Eq. (3). Denote

$$k = \frac{NC_2}{NC_1}. \tag{5}$$

The orbit of the Apollonian circle O_2 can be expressed as

$$\left(x - \frac{k+1}{k-1}c\right)^2 + y^2 = \frac{4k}{(k-1)^2}c^2. \tag{6}$$

Taking the part that intersects with the circumference of circle O_2 in the range of circle O_1, the possible trajectory of point N at t_1 is obtained.

We assume that $NC_2 > NC_1$, $k > 1$ because $\frac{NC_2}{NC_1}$ and $\frac{NC_1}{NC_2}$ have symmetry. By Eq. (6), we can see that when k increases by Δk, the x-axis coordinate increment of the center of circle O_2 is $\frac{(k+1)c}{(k-1)\Delta k}$, the y-axis coordinate of the center of circle O_2 is kept at coordinate 0, and the increment of the radius of circle O_2 is $\frac{(k+1)c}{(k-1)\Delta k} + \frac{c}{\Delta k}$. Therefore, when k increases to k' and O_2 changes to O_2', O_2 completely wraps circle O_2', which means that the circle O_1 is tangent to the circle O_2 when k reaches its maximum; mark it as the red circle centered on the point M as shown in the figure. For simplification, we let

$$t = \frac{k+1}{k-1} \tag{7}$$

Because of $k > 1$, t decreases when k increases. So, to find $k_{\max}$, we need to find $t_{\min}$. We have

$$\sqrt{(x_1 - ct_{\min})^2 + {y_1}^2} = r + c\sqrt{{t_{\min}}^2 - 1}. \tag{8}$$

By simplifying Eq. (8), we get

$$At_{\min}^2 + Bt_{\min} + C = 0 \tag{9}$$

$$A = 4x_1^2c^2 - 4r^2c^2 \tag{10}$$

$$B = -4x_1c\left({x_1}^2 + {y_1}^2 + c^2 - r^2\right) \tag{11}$$

$$C = \left({x_1}^2 + {y_1}^2 + c^2 - r^2\right)^2 + 4r^2c^2. \tag{12}$$

By solving Eq. (9), we get

$$t_{\min} = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \tag{13}$$

$$k_{\max} = \frac{t_{\min} + 1}{t_{\min} - 1}. \tag{14}$$

When $0 < k < 1$, we have

$$k_{\min} = \left(\frac{NC_2}{NC_1}\right)_{\min} = \frac{1}{\left(\frac{NC_1}{NC_2}\right)_{\max}}. \tag{15}$$

By symmetry and according to Eq. (14), $\left(\frac{NC_1}{NC_2}\right)_{\max} = k_{\max}$, so we have

$$k_{\min} = \frac{t_{\min} - 1}{t_{\min} + 1}. \tag{16}$$

From the above, the range of I can be calculated:

$$I = \left[\ \frac{-2A-B+\sqrt{B^2-4AC}}{2A-B+\sqrt{B^2-4AC}},\ \frac{2A-B+\sqrt{B^2-4AC}}{-2A-B+\sqrt{B^2-4AC}}\ \right]. \tag{17}$$

4.3 Location prediction

The range of the interval I derived above is derived from some certain values (x, y, c, t) and one uncertain value v. Note that not all member nodes keep the maximum speed at all moments, there is a certain law in the movement of member nodes over a period of time. Meanwhile, we can decrease v to decrease the FPR. But when v decreases, the TPR will also decrease. To increase the accuracy of the algorithm, we introduce an intelligent

algorithm to predict the movement of member nodes, so that we can appropriately reduce the interval I and improve the accuracy of the algorithm judgment.

At present, the position prediction algorithm has been more mature, such as work [38–42]. Considering the accuracy of prediction and the complexity of the algorithm for edge nodes, we use the long short-term memory (LSTM) model to predict the position of u_i based on the historical position of u_i, and we call it point $N_p(x_p, y_p)$. To make our training model more robust, we can use generative adversarial networks (GANs) [43] to train the model. By the way, to facilitate the development of intelligent algorithms for edge nodes, we rely on the employment of mature application programming interfaces (APIs) to standardize the algorithm. Work [44] proposed a collaborative framework for APIs recommendation.

According to Eq. (3), we could calculate η_p:

$$\eta_p = \left(\frac{(x_p + c)^2 + y_p^2}{(x_p - c)^2 + y_p^2} \right)^{\alpha}. \tag{18}$$

We use a maximum speed $v_{\max}$ to control the maximum range of activity, and a safe speed $v_{\min}$, which is regarded as a reasonable interval if the η_p is in the interval I calculated by $v_{\min}$. Then, we can calculate two intervals $I_{\min} = [\eta_{\min}, \eta'_{\min}]$ and $I_{\max} = [\eta_{\max}, \eta'_{\max}]$ using $v_{\min}$ and $v_{\max}$ according to Eq. (17). Because the path we predict is a point, we can expand its scope to increase robustness:

$$\mathrm{sub}_p = \max \left(\eta_{\max}, \frac{\eta_{\max} + w\eta_p}{1 + w} \right) \tag{19}$$

$$\mathrm{sup}_p = \min \left(\eta'_{\max}, \frac{\eta'_{\max} + w\eta_p}{1 + w} \right) \tag{20}$$

where w is the weight of η_p in the weighted average. Since $I_{\min} \subset I_{\max}$, we can get the prediction interval P:

$$P = \left[\min(\mathrm{sub}_p, \eta_{\min}), \max(\mathrm{sup}_p, \eta'_{\min}) \right] \tag{21}$$

Algorithm 1 Sybil attack detection algorithm

for each u_i in V_u **do**
  for each e_j in V_e **do**
    Calculate $d_i{}^j$ between u_i and e_j(by Eq. (2))
    Pick out two nearest edge nodes e_{n_1} and e_{n_2}
  end for
  At time t_0
  u_i sends a control package to e_{n_1} to e_{n_2} respectively
  e_{n_1} calculates $R_1{}^{i0}$ (by Eq. (1))
  e_{n_2} calculates $R_2{}^{i0}$ (by Eq. (1))
  e_{n_1} send a package including $R_1{}^{i0}$ to e_{n_2}
  e_{n_2} calculates η_0 (by Eq. (3))
  e_{n_2} calculates interval I_{min} and I_{max} (by Eq. (17))
  e_{n_2} calculates η_p and P (by Eq. (21))
  At time t_1**, repeat line 7-10**
  e_{n_2} calculates η_1 (by Eq. (3))
  while not all packages have been handled **do**
    if e_2 received two same ids **then**
      if η_2 in P **then**
        Return normal node
      else
        Return Sybil node
      end if
    else
      if η_2 in P **then**
        Return Sybil node
      else
        Return normal node
      end if
    end if
  end while
end for

4.4 Performance evaluation

In this section, the performance evaluation is proposed from three aspects: computation, memory and communication. For two edge nodes e_i and e_j, we delete that the number of member nodes connected to both of them is M.

The computational complexity is evaluated as follows. Because the detection cycle of the algorithm is two rounds, we only evaluate in the two rounds. In each round, two edge nodes are subject to a control packet of one member node within their jurisdiction. The edge nodes take $O(2M)$ time to calculate the RSSI value and take $O(M)$ time for average to calculate the rate of RSSI pairs. In the first round, $O(M)$ time is taken for calculate the interval P. In the second round, the pairing of nodes takes $O(M \log M)$ time. Therefore, the time complexity of one edge node is $\left(O\left(\frac{2M+M+M+M\log M}{2}\right)\right) = O(M \log M)$. Consider that when edge nodes' number increases, M decreases, then each edge node consume less computation.

The following are the statistics of memory overhead. In each round, two edge nodes are subject to a control packet of one member node within their jurisdiction. Assuming that the size of each packet is S, then e_i and e_j occupy a total of $4SM$ memory when storing control packets in the whole scheme because memory is emptied after every two rounds.

Table 1 Notations for problem statement

u_i	A normal member node or a malicious node
e_j	The j-th edge node
V	The set of all nodes
V_n	Normal nodes set
V_m	Malicious nodes set
V_u	$V_n \cup V_m$
V_e	Edge nodes set
d_j^i	Distance between node u_i and edge node e_j
R_j^{ir}	RSSI value from u_i to e_j in the r-th round, r=1, 2
η_r	The ratio got in the r-th round, r=1, 2
v_i	The speed of u_i

In the location-predicting step, we need to use the historical data of multiple groups of nodes to predict the location, so klM bits of memory in this step is taken, where k is the historical location number of each node and l is the bit size of each location. In addition, at the end of the first round, e_i and e_j take a total of $|P|M$ bits to store the interval P, where $|P|$ is the bit size of one interval datum. Therefore, the average memory cost one edge nodes is $\frac{(4S+kl+|P|)M}{2} = O(M)$. Consider that when edge nodes' number increases, M decreases, then each edge node consume less memory. Each member node has no memory overhead in the whole scheme. Generally speaking, the memory overhead generated by this algorithm is very small.

Communication overhead of our method is also evaluated. The communication overhead is also evaluated. Because of the energy limitation of sensor nodes, IoT algorithms need to pay attention to energy consumption. For sensor nodes, the energy consumption of transmitting information wirelessly is much more than computation, so we focus on the analysis of the transmitted packets. In a whole detection cycle, each member node send 2 packages to edge nodes who are connected to it. Meanwhile, the data transmission of e_i and e_j is $2M$, so the data transmission of each edge node is $2M + \frac{2M}{2} = 3M$ and is proportional to the number of nodes connected to it.

5 Experimental evaluation and summary of experiments

In this section, the performance of the algorithm is evaluated through a series of experiments coded in python. We evaluate our experiments from 2 metrics including TPR and FPR (Table 1).

5.1 Experimental evaluation

We designed three independent experiments to evaluate this method. In the experiments, we simulate three indicators: the number of edge nodes (E), the number of normal nodes(N) and the number of Sybil nodes(S). All the edge nodes and randomly distributed member nodes are located in a square field with a side length of 100 m. We repeated each experiment 200 times and analyzed its average. The parameters related to the experiment are shown in Table 2.

The first experiment shows how N affects TPR and FPR. In this experiment, we fix the parameters $S = 40$ and $E = 8$ while increasing N from 100 to 500. We also study the effect of the number of execution rounds: The parameter R is varied in increasing

Table 2 Experimental parameters

Parameters	Values
Network size	$100 \times 100\,m^2$
Number of normal nodes	$N = 100, 200, 300, 400, 500$
Number of edge nodes	$E = 4, 8, 12$
Number of Sybil nodes	$S = 10, 20, 30, 40$
Number of monitoring rounds	$R = 20, 40, 60, 80, 100$
Speed of a member node	$v_i \in [0, 2]\,m/s$
Safe speed	$v_{max} = 2\,m/s$
Min speed	$v_{min} = 0.3\,m/s$
Communication range between member nodes	[0, 50] m

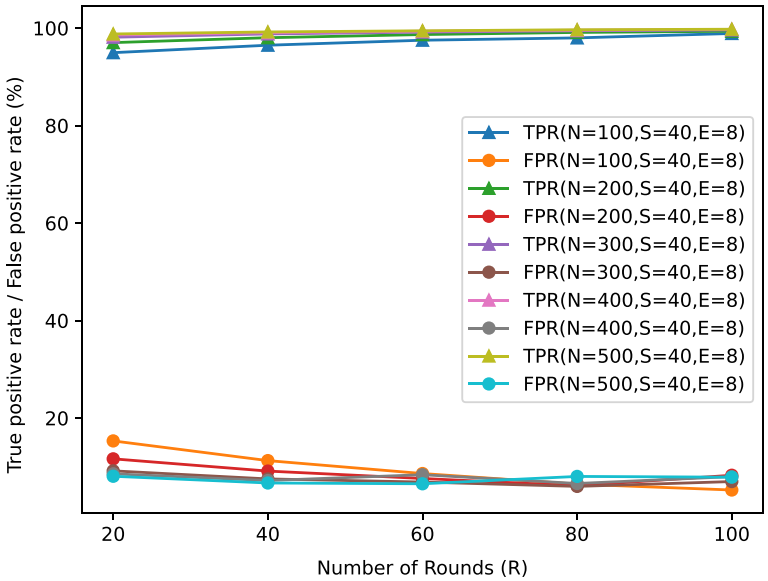

Fig. 4 Effects of the number of normal nodes N on detection performance

steps of 20 in the range of 20–100, and in each round of the experiment, 1/3 Sybil nodes are randomly selected to attack, which makes the experiment closer to the real environment. After each round, the system kicks off the suspicious nodes. The results are shown in Fig. 4. The experiment shows the TPR increases as N increases, and the TPR exceeds 94% for $N = 100$. As mentioned before, if an id conflict is found in the detection, the edge node will resort to base stations, which can verify nodes' identity and thus improve the TPR. Another reason of the high TPR is that the η_2 value belonging to interval P is a basic assumption in this algorithm to judge whether a node is a normal node. Furthermore, the ratio of normal nodes to Sybil nodes has a significant impact on FPR according the results. In the first round, the FPR decreases slightly as N increases. In the network, when there are fewer normal nodes, Sybil attackers may pretend to be nearby nodes who are also Sybil attackers, so that the attack goes undetected. At the same time, when there are fewer Sybil nodes, misclassified normal nodes increase the FPR sharply, so after the first round, the FPR fluctuates greatly.

Table 3 Comparison of performance of the proposed algorithm and other latest algorithms in terms of TPR

Algorithm	TPR (%)
Gandino et al. [45]	99
Jamshidi et al. [24]	99
Jamshidi et al. [14]	94
Yao et al. [30]	90
Garip et al. [29]	87
Proposed algorithm	94

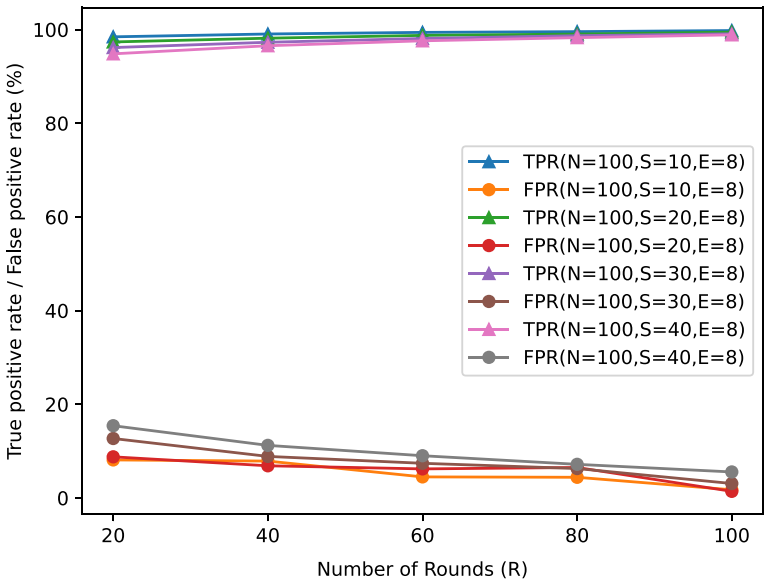

Fig. 5 Effects of the number of Sybil nodes S on detection performance

In addition, we compare the TPR performance of our scheme with other state-of-the-art schemes. The TPR of the algorithms proposed by Gandino et al. [45], Jamshidi et al. [24], Jamshidi et al. [14], Yao et al. [30] and Garip et al. [29] are about to be 99%, 99%, 94%, 90% and 87%, respectively, as shown in Table 3. Some schemes perform better than the proposed scheme, while the overhead in the aspects of computation, communication and memory of our scheme is lower. So, the experimental results are consistent with the expected performance of our scheme.

We study the effect of S in the second experiment. In this experiment, we fix the parameters $N = 100$ and $E = 8$ while increasing S from 10 to 40. We also study the effect of the number of execution rounds: The parameter R is varied in increasing steps of 20 in the range of 20–100, and in each round of the experiment, 1/3 Sybil nodes are randomly selected to attack. As shown in Fig. 5, the results of the first and the second experiments are similar. The TPR decreases as S increases, and the FPR increases slightly as S increases.

We study the effect of E in the third experiment. In this experiment, we fix the parameters $N = 100$ and $S = 40$, $E = 4, 8, 12$. We also study the effect of the number

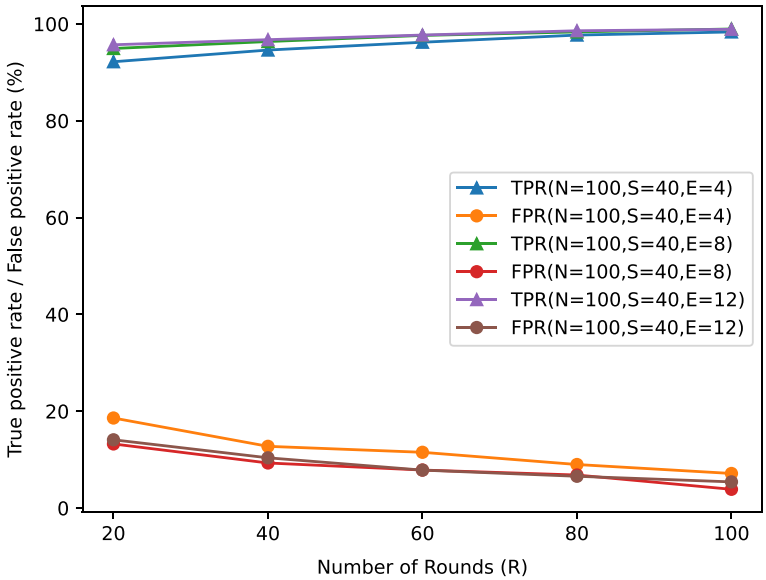

Fig. 6 Effects of the number of edge nodes C on detection performance

Table 4 Summary of experimental results

Parameter↑	True positive rate	False positive rate
N	↑	↓
S	↓	↑
E	↑	↓

↑: Increase

↓: Decrease

↕: Uncertain

of execution rounds: The parameter R is varied in increasing steps of 20 in the range of 20 to 100, and in each round of the experiment, 1/3 Sybil nodes are randomly selected to attack. As shown in Fig. 6, the TPR results of the above three experiments are similar, the TPR increases as E increases. The FPR decreases as E increases. In particular, when $E = 4$, the FPR is slightly worse. One explanation is that when there are few edge nodes, the angle between one member node and two edge nodes may be small, which it is easy to cause errors in calculating I.

5.2 Summary of the experiments

Our proposed algorithm achieves more than 94% TPR and less than 14% FPR when the number of edge nodes is greater than 4, and it also achieves more than 92% TPR and less than 16% FPR when the number of edge nodes equals 4. Some conclusions in terms of parameters in the experiments are summarized in Table 4.

6 Conclusion

In this paper, a lightweight mobile IoT Sybil attack detection method based on edge computing is proposed. An important contribution of this paper is strict mathematical verification used to find out the member nodes' feature's feasible fluctuation interval, and intelligent algorithms are used to predict the position of member nodes, which makes the interval more accurate. In theory, our scheme has a small overhead in aspects of computation, communication and memory. The experimental results show the scheme has good performance in mobile IoT. The FPR of our scheme is relatively susceptible which is one of the shortcomings, which can be improved by improving the algorithm in the future work.

Abbreviations

AOA	Angle of arrival
API	Application programming interface
CA	Code attestation
DDoS	Distributed denial of service
DNS	Domain name system
FPR	False positive rate
GAN	Generative adversarial network
GPS	Global positioning system
IoT	Internet of things
LSTM	Long short-term memory
p2p	Peer-to-peer
PV	Position verification
RKP	Random key pre-distribution
RPL	Routing protocol for low-power and lossy networks
RRT	Radio resource test
RSSI	Received signal strength indication
RSU	Road side unit
TDOA	Time difference of arrival
TPR	True positive rate
VANET	Vehicular ad hoc network
WSN	Wireless sensor network

Author information

Junwei Yan received a bachelor's degree in computer science from Shanghai Jiao Tong University (SJTU), China, in 2017. He is currently studying for a master's degree at the Department of Computer Science and Engineering, Shanghai Jiaotong University.

Tao Jiang received a bachelor's degree in Measurement and Control Technology and Instruments from China University of Geosciences (CUG) in 2004. He obtained a master's degree in 2015, majoring in Precision machinery and Instrumentation at Northwestern Polytechnical University, China. In recent years, he is mainly engaged in computer simulation, measurement and control data processing and other directions.

Liwei Lin received a Ph.D. degree in computer science from Shanghai Jiao Tong University (SJTU), China, in 2020. He is now a lecturer at School of Computer Science and Mathematics, Fujian University of Technology, China. His research interests include data center network, mobile computing and cloud computing.

Zhengyu Wu received a BS degree in software engineering from Shanghai Jiao Tong University (SJTU), China, in 2020. He serves as a system engineer at Honor Inc. currently. His research interests include cloud computing, mobile computing and artificial intelligence.

Xiucai Ye received her Ph.D. in computer science from University of Tsukuba, Tsukuba Science City, Japan, in 2014. She is currently working as an assistant professor at the Department of Computer Science and Center for Artificial Intelligence Research (C-AIR), University of Tsukuba, Tsukuba Science City, Japan. Her current research interests include clustering, feature selection, machine learning and bioinformatics. She is a member of IEEE.

Mengke Tian is working for Beijing Institute of Technology as an Engineer. At the same time, she is doing postdoctoral research in the Institute of Microelectronics, Peking University. She graduated with a BE degree in Automation technology from Huazhong University of Science and Technology (2014) and earned her MPhil degree (2016) and Ph.D. degree (2019) in Industrial Engineering and Decision Analytics from the Hong Kong University of Science and Technology.

Yong Wang received a Ph.D. degree in electronic encapsulating technology from Beijing Institute of Technology, China. He is the head of Beijing Microelectronics Technology Institute. He is a technical leader of high reliability packaging technology for aerospace.

Acknowledgements

This work was supported in part by the National NSF of China (Nos. 61872234, 61732010 and 62172095), Shanghai Key Laboratory of Scalable Computing and Systems, Educational scientific research project of Fujian Provincial Department

of Education (No. JAT210291), the Science Foundation of Fujian University of Technology (No. GY-Z220206), Innovative Research Foundation of Ship General Performance (No. 25622114) and the Key Laboratory of PK System Technologies Research of Hainan.

Author contributions
All authors read and approved the final manuscript.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 5 September 2022 Accepted: 6 February 2023
Published online: 05 March 2023

References

1. L. Tan, N. Wang, Future internet: the internet of things. In *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, vol. 5 (IEEE, 2010), pp. 5–376
2. A. Whitmore, A. Agarwal, L. Da Xu, The internet of things—a survey of topics and trends. Inf. Syst. Front. **17**(2), 261–274 (2015). https://doi.org/10.1007/s10796-014-9489-2
3. L. Da Xu, W. He, S. Li, Internet of things in industries: a survey. IEEE Trans. Ind. Inf. **10**(4), 2233–2243 (2014)
4. R. Zhang, X. Chu, R. Ma, M. Zhang, L. Lin, H. Gao, H. Guan: OSTTD: Offloading of splittable tasks with topological dependence in multi-tier computing networks. IEEE J. Sel. Areas Commun. JSAC (2022)
5. M. Chen, S. Mao, Y. Liu, Big data: a survey. Mob. Netw. Appl. **19**(2), 171–209 (2014)
6. J.R. Douceur, The Sybil attack. In *International Workshop on Peer-to-Peer Systems* (Springer, 2002), pp. 251–260
7. B. Parno, A. Perrig, Challenges in securing vehicular networks. In *Workshop on Hot Topics in Networks (HotNets-IV)* (Maryland, USA, 2005), pp. 1–6
8. M. Wen, H. Li, Y.-F. Zheng, K.-F. Chen, TDOA-based Sybil attack detection scheme for wireless sensor networks. J. Shanghai Univ. (Engl. Edn.) **12**(1), 66–70 (2008)
9. K.-F. Ssu, W.-T. Wang, W.-C. Chang, Detecting Sybil attacks in wireless sensor networks using neighboring information. Comput. Netw. **53**(18), 3042–3056 (2009)
10. M. Demirbas, Y. Song, An RSSI-based scheme for Sybil attack detection in wireless sensor networks. In *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks* (IEEE Computer Society, 2006), pp. 564–570
11. J. Newsome, E. Shi, D. Song, A. Perrig, The Sybil attack in sensor networks: analysis and defenses. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks* (ACM, 2004), pp. 259–268
12. Y. Zhang, K. Fan, S. Zhang, W. Mo, AOA based trust evaluation scheme for Sybil attack detection in WSN. Appl. Res. Comput. **27**(5), 1847–1849 (2010)
13. C. Piro, C. Shields, B.N. Levine, Detecting the Sybil attack in mobile ad hoc networks. In *Securecomm and Workshops* (IEEE, 2006), pp. 1–11
14. M. Jamshidi, E. Zangeneh, M. Esnaashari, M.R. Meybodi, A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks. Comput. Electric. Eng. **64**, 220–232 (2017)
15. L. Lin, D.S. Wei, R. Ma, J. Li, H. Guan, Online traffic-aware linked VM placement in cloud data centers. Sci. China Inf. Sci. **63**(7), 1–23 (2020)
16. R. Ma, J. Li, H. Guan, M. Xia, X. Liu, EnDAS: efficient encrypted data search as a mobile cloud service. IEEE Trans. Emerging Top. Comput. **3**(3), 372–383 (2015)
17. Z. Qi, C. Xiang, R. Ma, J. Li, H. Guan, D.S. Wei, Forenvisor: A tool for acquiring and preserving reliable data in cloud live forensics. IEEE Trans. Cloud Comput. **5**(3), 443–456 (2017)
18. H.-J. Shao, X.-P. Zhang, Z. Wang, Efficient closed-form algorithms for AOA based self-localization of sensor nodes using auxiliary variables. IEEE Trans. Signal Process. **62**(10), 2580–2594 (2014)
19. S. Zhong, L. Li, Y.G. Liu, Y.R. Yang, *Privacy-Preserving Location-based Services for Mobile Users in Wireless Networks*. Department of Computer Science, Yale University, Technical Report ALEU/DCS/TR-1297 (2004)
20. J. Wang, G. Yang, Y. Sun, S. Chen, Sybil attack detection based on RSSI for wireless sensor network. In *International Conference on Wireless Communications, Networking and Mobile Computing, WiCom 2007* (IEEE, 2007), pp. 2684–2687
21. A.O. Bang, U.P. Rao, A novel decentralized security architecture against Sybil attack in RPL-based IoT networks: a focus on smart home use case. J. Supercomput. **77**(12), 13703–13738 (2021). https://doi.org/10.1007/s11227-021-03816-2
22. H. Guo, H. Wang, T. Song, Y. Hua, Z. Lv, X. Jin, Z. Xue, R. Ma, H. Guan, Siren: Byzantine-robust federated learning via proactive alarming. In *ACM Symposium on Cloud Computing* (2021), pp. 47–60
23. J. Zhang, Y. Hua, T. Song, H. Wang, Z. Xue, R. Ma, H. Guan, Improving Bayesian neural networks by adversarial sampling. AAAI (2022)
24. M. Jamshidi, M. Ranjbari, M. Esnaashari, N.N. Qader, Sybil node detection in mobile wireless sensor networks using observer nodes. JOIV: Int. J. Inform. Vis. **2**(3), 159–165 (2018)
25. R. Shyamala, S. Valli, Impact of blackhole and rushing attack on the location-based routing protocol for wireless sensor networks, in *Advances in Computing and Information Technology*. ed. by N. Meghanathan, D. Nagamalai, N. Chaki (Springer, Berlin, 2012), pp.349–359
26. R. Muraleedharan, X. Ye, L.A. Osadciw, Prediction of Sybil attack on WSN using Bayesian network and swarm intelligence. In *Wireless Sensing and Processing III*, vol. 6980 (International Society for Optics and Photonics, 2008), p. 69800

27. S. Sharmila, G. Umamaheswari, Detection of Sybil attack in mobile wireless sensor networks. Int. J. Eng. Sci. Adv. Technol. **2**(2), 256–262 (2012)
28. D.L.S.S. Reddy, V. Bapuji, A. Sarma, S.S.V.N. Sarma, Sybil attack detection technique using session key certificate in vehicular ad hoc networks. In *2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET)* (2017), pp. 1–5
29. M.T. Garip, P.H. Kim, P. Reiher, M. Gerla, Interloc: An interference-aware RSSI-based localization and Sybil attack detection mechanism for vehicular ad hoc networks. In *2017 14th IEEE AnnualConsumer Communications and Networking Conference (CCNC)* (IEEE, 2017), pp. 1–6
30. Y. Yao, B. Xiao, G. Wu, X. Liu, Z. Yu, K. Zhang, X. Zhou, Multi-channel based Sybil attack detection in vehicular ad hoc networks using RSSI. IEEE Trans. Mob. Comput. (2018)
31. H. Gao, B. Qiu, R.J.D. Barroso, W. Hussain, Y. Xu, X. Wang, Tsmae: a novel anomaly detection approach for internet of things time series data using memory-augmented autoencoder. IEEE Trans. Netw. Sci. Eng. (TNSE) (2022). https://doi.org/10.1109/TNSE.2022.3163144
32. K. Akkaya, M. Younis, A survey on routing protocols for wireless sensor networks. Ad Hoc Netw. **3**(3), 325–349 (2005)
33. A. Savvides, C.-C. Han, M.B. Strivastava, Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking* (ACM, 2001), pp. 166–179
34. H. Gao, W. Huang, T. Liu, Y. Yin, Y. Li, Ppo2: Location privacy-oriented task offloading to edge computing using reinforcement learning for intelligent autonomous transport systems. IEEE Trans. Intell. Transp. Syst. 1–14 (2022). https://doi.org/10.1109/TITS.2022.3169421
35. W.C. Jakes, D.C. Cox, *Microwave Mobile Communications* (Wiley-IEEE Press, 111 River Street, Hoboken, NJ, USA, 1994)
36. N.W. Lo, D.D. Falconer, A.U. Sheikh, Adaptive equalization and diversity combining for mobile radio using interpolated channel estimates. IEEE Trans. Veh. Technol. **40**(3), 636–645 (1991)
37. J.L. Burbank, W. Kasch, J. Ward, *An Introduction to Network Modeling and Simulation for the Practicing Engineer*, vol. 5 (Wiley, 111 River Street, Hoboken, NJ, USA, 2011)
38. R.W. Liu, M. Liang, J. Nie, W.Y.B. Lim, Y. Zhang, M. Guizani, Deep learning—powered vessel trajectory prediction for improving smart traffic services in maritime internet of things. IEEE Trans. Netw. Sci. Eng. **9**(5), 3080–3094 (2022). https://doi.org/10.1109/TNSE.2022.3140529
39. X. Wan, H. Liu, H. Xu, X. Zhang, Network traffic prediction based on LSTM and transfer learning. IEEE Access **10**, 86181–86190 (2022). https://doi.org/10.1109/ACCESS.2022.3199372
40. R. Quan, L. Zhu, Y. Wu, Y. Yang, Holistic LSTM for pedestrian trajectory prediction. IEEE Trans. Image Process. **30**, 3229–3239 (2021). https://doi.org/10.1109/TIP.2021.3058599
41. L. Lin, W. Li, H. Bi, L. Qin, Vehicle trajectory prediction using LSTMs with spatial–temporal attention mechanisms. IEEE Intell. Transp. Syst. Mag. **14**(2), 197–208 (2022). https://doi.org/10.1109/MITS.2021.3049404
42. H. Xue, D.Q. Huynh, M. Reynolds, Poppl: Pedestrian trajectory prediction by LSTM with automatic route class clustering. IEEE Trans. Neural Netw. Learn. Syst. **32**(1), 77–90 (2021). https://doi.org/10.1109/TNNLS.2020.2975837
43. H. Gao, B. Dai, H. Miao, X. Yang, R.J.D. Barroso, H. Walayat, A novel gapg approach to automatic property generation for formal verification: The gan perspective. ACM Trans. Multimed. Comput. Commun. Appl. (2022). https://doi.org/10.1145/3517154
44. Y. Xu, Y. Wu, H. Gao, S. Song, Y. Yin, X. Xiao, Collaborative APIs recommendation for artificial intelligence of things with information fusion. Future Gener. Comput. Syst. **125**, 471–479 (2021). https://doi.org/10.1016/j.future.2021.07.004
45. F. Gandino, R. Ferrero, M. Rebaudengo, A key distribution scheme for mobile wireless sensor networks: *q-s*-composite. IEEE Trans. Inf. Forensics Secur. **12**(1), 34–47 (2017)

Publisher's Note