EMBRY-RIDDLE
Aeronautical University
SCHOLARLY COMMONS

Publications

7-20-2019

Adaboost-Based Security Level Classifcation of Mobile Intelligent Terminals

Feng Wang
University of Electronic Science and Technology of China

houbing song
Embry-Riddle Aeronautical University, SONG4@erau.edu

Dingde Jiang
University of Electronic Science and Technology of China, jiangdd@uestc.edu.cn

Hong Wen
University of Electronic Science and Technology of China, 70000413@qq.com

Scholarly Commons Citation

Wang, F., song, h., Jiang, D., & Wen, H. (2019). Adaboost-Based Security Level Classifcation of Mobile Intelligent Terminals. *The Journal of Supercomputing,* (). https://doi.org/10.1007/s11227-019-02954-y

The Journal of Supercomputing
https://doi.org/10.1007/s11227-019-02954-y

Adaboost-based security level classification of mobile intelligent terminals

Feng Wang[1] · Dingde Jiang[1] · Hong Wen[1] · Houbing Song[2]

Abstract

With the rapid development of Internet of Things, massive mobile intelligent terminals are ready to access edge servers for real-time data calculation and interaction. However, the risk of private data leakage follows simultaneously. As the administrator of all intelligent terminals in a region, the edge server needs to clarify the ability of the managed intelligent terminals to defend against malicious attacks. Therefore, the security level classification for mobile intelligent terminals before accessing the network is indispensable. In this paper, we firstly propose a safety assessment method to detect the weakness of mobile intelligent terminals. Secondly, we match the evaluation results to the security level. Finally, a scheme of security level classification for mobile intelligent terminals based on Adaboost algorithm is proposed. The experimental results demonstrate that compared to a baseline that statistically calculates the security level, the proposed method can complete the security level classification with lower latency and high accuracy when massive mobile intelligent terminals access the network at the same time.

Keywords Internet of Things · Adaboost · Edge server · Mobile intelligent terminal · Security level classification

✉ Houbing Song
h.song@ieee.org

Dingde Jiang
jiangdd@uestc.edu.cn

Hong Wen
70000413@qq.com

[1] School of Astronautics and Aeronautic, University of Electronic Science and Technology of China, Chengdu 611731, China

[2] Department of Electrical Computer, Software, and Systems Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, USA

Published online: 20 July 2019

1 Introduction

Mobile intelligent terminals are becoming the main tool to obtain information in people's daily life [1]. According to the ITU-T study, more than 50 billion devices will access the Internet by 2020, and each person will generate 1.7 MB of data per second on average [2]. With the increasingly powerful third-party applications that are closely related to people's privacy, such as online shopping, mobile banking and chatting applications, we may inadvertently store the property information, personal privacy, trade secret documents and other intimate information in mobile intelligent terminals, thus resulting in potential risks of privacy leakage [3–5]. Meanwhile, with the advent of the Internet of Things era and the transition from cloud computing to edge computing architecture, massive mobile intelligent terminals may carry out data interaction under the self-organizing network at any time [6, 7]. For example, various types of terminals in a public building can process, transfer and share data in real time through an edge server which is built nearby [8]. In addition to achieving low latency and centralized data processing, the edge server faces new challenges in terminals management and privacy protection of datasets [9, 10]. Therefore, it is necessary to classify the security level of mobile intelligent terminals before accessing the network [11].

The main purpose of this paper is to realize fast security level classification for mobile intelligent terminals. The security level classification is one of the most effective means to ensure the management and safe handling of terminals, and the concept is shown in Fig. 1. Before mobile intelligent terminals access the network, the edge server first conducts security assessment on the items where data interaction may occur. Then, according to the testing results of each item, the scientific

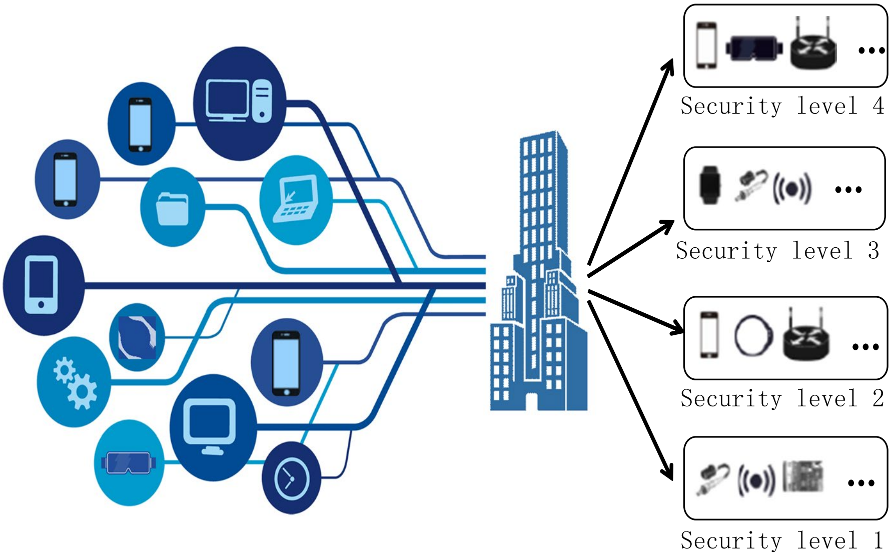


Fig. 1 Classification concept of security levels of IoT mobile intelligent terminals

terminal safety level is graded, so that the edge server has a clear understanding of the capabilities of various terminals against attacks. The classification of security level is an important safety-employed index to different organizations, and it will assist the edge server realizing the reasonable dispatch of mobile intelligent terminals [12].

How to carry out a comprehensive security assessment for mobile intelligent terminals and design a classification algorithm with high precision and low complexity is the main challenge [13, 14]. Cavalcanti et al. [15] proposed that Android-based terminals can identify and report potential unsafe settings by comparing pre-defined risk lists when terminals accessed the Internet in an insecure environment. This method provided a detailed analysis of possible security flaw in intelligent terminals, but no further operation was given for safety use. Irwan et al. [16] analyzed the latent security vulnerabilities in intelligent terminals and found the relevant permissions or configuration parameters so that they could enhance terminal security by strengthening the management of related permissions. However, the proposed assessment model was relatively simple. Authors only considered the permission or configuration changes made by the applications in the terminal and didn't analyze whether there were security vulnerabilities in the operating system.

At present, researchers often use some supervised learning models to realize fast and accurate multidimensional data classification, such as back-propagation neural network (BPNN), support vector machines (SVM) and adaptive boosting (Adaboost). Dong et al. [17] used the semi-supervised convolutional neural network to classify vehicle types so that it could automatically learn better characteristic parameters for classification in complex scenarios. Deng et al. [18] attempted to combine fuzzy learning algorithm with neural networks in order to solve the problem of high uncertainty data classification. Guo et al. [19] proposed a SVM-based sequential classifier training (SCT-SVM) approach that applied progressively to sequential image data for multitemporal remote sensing image classification. Ding et al. [20] utilized Adaboost algorithm for dynamic gesture recognition and achieved high accuracy. According to these researches, although all three algorithms possess excellent classification function, the resource consumption of model training including argument numbers, layer number and sample quantity is different. The choice of algorithm impacts both the resource consumption and latency of the security level classification. For example, using NN models with more layers or SVM models operated in higher dimensional space enables classification accurate but also demands more GPU processing. The processing capability of edge servers can't afford such a large resource demand when massive terminals access simultaneously, which will affect the model initialization and update as well as time latency. The best classification scheme is the one with the lowest resource demand whose accuracy is over the desired threshold. Considering that the computing capacity of the edge server is not outstanding, it is most effective to select Adaboost, a simple but efficient algorithm to classify the security level of mobile intelligent terminal. Our previous work can be found in [21–24].

This paper summarizes the deficiency of the above research. Firstly, a scientific security assessment method is designed to detect the security vulnerabilities of mobile intelligent terminals. Then, different security levels are matched with the testing results by the statistical approach. Finally, a classification method based on the Adaboost algorithm is proposed to realize not only high-precision security level division, but the rapid data processing with massive mobile intelligent terminals accessing the network simultaneously.

In conclusion, the main contribution of this paper is expressed as follows:

- We propose a security level assessment scheme for terminals, through 22 test items and a complete test process to achieve security assessment.
- An Adaboost-based security level classification algorithm is proposed for mobile intelligent terminals to realize fast and precise classification.

The rest of this paper is organized as follows. Section 2 proposes an assessment method for terminal security. An Adaboost-based model for security level classification of mobile intelligent terminals is established in Sect. 3. Section 4 presents the experimental results and analysis. Finally, we conclude our work in Sect. 5.

2 Security assessment of mobile intelligent terminals

As shown in Fig. 2, this section sets up 22 test items based on common security vulnerabilities and puts forward the security evaluation standards of each item to achieve the comprehensive security evaluation of mobile intelligent terminals. Referring to ISO/IEC 25040-2011 standards, there are two kinds of security assessment results of each test item in mobile intelligent terminal:

- No abnormalities: no safety risk or incident is found through the assessment method;

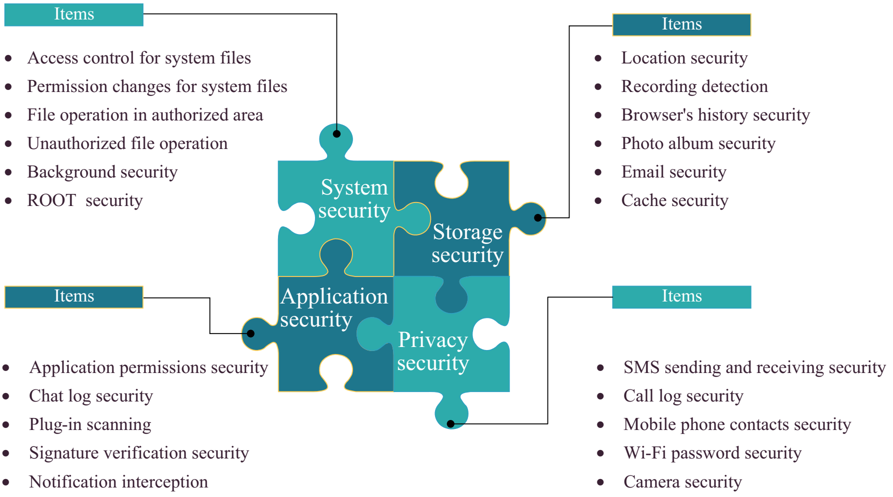

Items
Access control for system files
Permission changes for system files
File operation in authorized area
Unauthorized file operation
Background security
ROOT security
Items
Location security
Recording detection
Browser's history security
Photo album security
Email security
Cache security
System security
Storage security
Application security
Privacy security
Items
Application permissions security
Chat log security
Plug-in scanning
Signature verification security
Notification interception
Items
SMS sending and receiving security
Call log security
Mobile phone contacts security
Wi-Fi password security
Camera security

Fig. 2 Security assessment items of mobile intelligent terminals

- Disqualification: directly discover incidents or fail to achieve security requirements.

In order to standardize the testing process of each item, taking the location security item as an example, the safety assessment steps and evaluation results are as follows:

Step 1 Check whether the operating system of the mobile intelligent terminal provides the development module of location function;
Step 2 If a development module of location function is provided by the operating system, detect the location function applications;
Step 3 Run the applications and check whether the terminal asks the user to confirm using location function.

The expected results are as follows:

After step 1, if the operating system does not provide the development module of location function, the evaluation result of this item is No abnormalities, and the evaluation ends;

After step 3, if asking the user to confirm, the evaluation result of this item is No abnormalities, and the evaluation ends;

After step 3, if the mobile intelligent terminal does not ask the user to confirm and invokes location function successfully, the evaluation result of this item is Disqualification, and the evaluation ends.

In the above, we have illustrated an example of location security test including the security assessment steps and evaluation results. We have "No abnormalities" corresponding to the value 1 and "Disqualification" corresponding to the value 0. For the other test items, the test process is similar. The testing process meets the requirements of ISO/IEC 25040-2011 standards. Each item should be tested, respectively, in the operating system, involved applications and relevant permissions. For each time of mobile intelligent terminal security assessment, we can get a testing result vector with 22 dimensions where the value of each dimension is 0 or 1.

3 Security level classification model

In this section, we firstly introduce the general methods and framework of terminal security classification. Then, the Adaboost-based model is proposed to realize fast classification of test data of terminals. The performance judgment of two methods is given in the end.

3.1 Framework

Before proposing the security level classification model based on Adaboost, we first introduce the basic model. As Fig. 3 shows, first of all, the mobile intelligent terminal is tested in 22 security testing items, and the testing results $[t_1, t_2, \dots, t_{22}]$

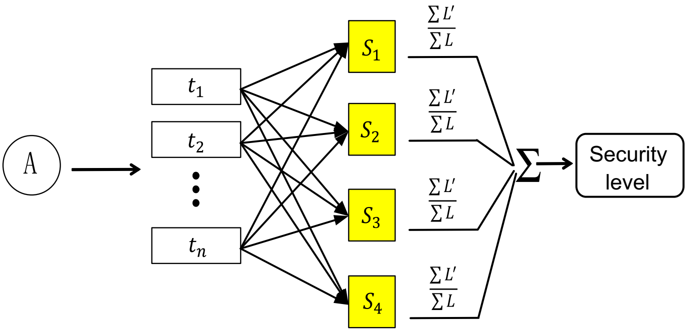


Fig. 3 A basic method for security level classification

can be obtained. According to the data flow in the terminal, we divide the data changes into four stages: collection, processing, transfer and deletion, which will reflect the data security in the terminal. Therefore, the purpose of all testing items is to test the security of one or more of the four stages for each terminal. The results of each stage are obtained by statistical analysis. After getting the testing result vector M_i, for each stage S_i, L is the total number of related test items and L' is the number of items tested in “No abnormalities.” We assume that the stage is safe as L'/L is greater than 0.8, otherwise the stage is insecure. The choice of 0.8 is not crucial, as Fig. 4 shows, we randomly compare the security levels of 20 terminals at different ratios and find the classification is dispersive better when L'/L is around 0.8, finally counting the number of safety stage to determine the security level of the terminal. In this paper, the terminal security is divided

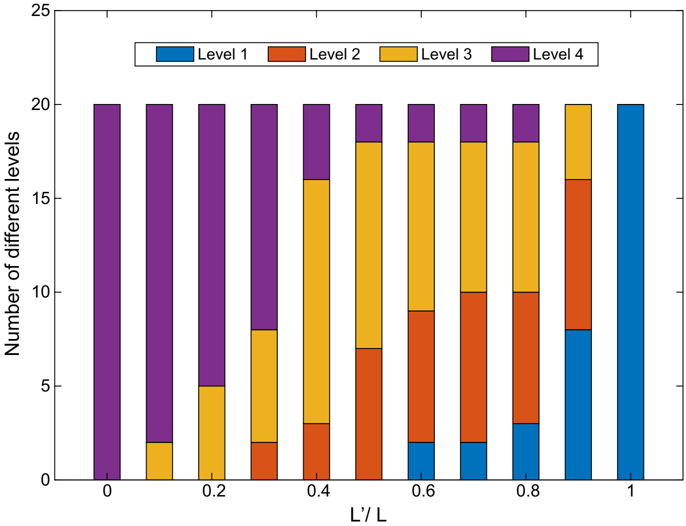


Fig. 4 Comparison of security levels at different ratios

into four levels: Worse, Normal, Good and Great, corresponding the tested safety stage number 0 or 1, 2, 3 and 4, as shown in Table 1.

For the calculation convenience, we, respectively, utilize Level 1, Level 2, Level 3 and Level 4 representing security levels Worse, Normal, Good and Great. During the test of one terminal, if only one or none of the four stages qualifies, then the security level is Level 1; if two stages qualify, then the security level is Level 2; if three stages qualify, then the security level is Level 3; and if four stages qualify, then the security level is Level 4. Higher levels reveal stronger safety condition.

The method above is the basic method for security level classification of mobile intelligent terminals, which is simple in process and accurate in result, but as a security assessment before accessing network, too much time and computing resources are wasted especially when massive terminals are accessing simultaneously under the IoT structure. So the purpose of this article is to seek a security level classification scheme for mobile intelligent terminals with low delay and high accuracy.

3.2 Adaboost-based model

AdaBoost is the abbreviation of adaptive boosting. It is a kind of dichotomy classification algorithm model, training a series of weak classifiers and combining them into a strong classifier to meet the classification requirements of datasets. The adaptation of AdaBoost is that the weight of the sample that is misclassified by the previous weak classifier will increase and the weight of the correctly classified sample will decrease to train next weak classifier. The final strong classifier is not determined until the error rate is small enough or the maximum number of iterations is reached. Different from the gradient boosting decision tree (GBDT) algorithm, the weak classifiers of Adaboost are independent of each other. However, there is a connection among the weak classifiers in GBDT, so additional parameter control functions are needed to prevent training errors. As a result, the Adaboost algorithm is selected for fast and precise data classification.

As shown in Fig. 5, the initial item testing is consistent with the basic method. Firstly, we test $N+P$ mobile intelligent terminals. Then the testing result $M_1, M_2, \dots, M_N, \dots, M_{N+P}$ is obtained. Each testing result consists of 22 testing items that is represented by the vector $M_i = [t_1, t_2, \dots, t_{22}]^T$, where t_i is the testing result of the testing item. Each testing result M_i is then preprocessed to obtain the testing result vector X_i and its security level y. Finally, we get the dataset T:

$$T = \{(X_1, y_1), (X_2, y_2), \dots, (X_{N+P}, y_{N+P})\} \tag{1}$$

where $X_i \in \chi \subseteq N^n$ and $y_i \in \{1, 2, 3, 4\}$. The first N sets of data are divided as the training set T_N, and the last P sets of data are used as the testing set T_P. Now we use

Table 1 Definition of the mobile intelligent terminal security level

Four data change stage	Collection	Processing	Transfer	Deletion
Safety stage number	0 or 1	2	3	4
Safety level	Worse	Normal	Good	Great

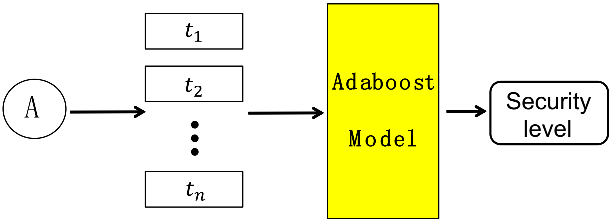


Fig. 5 Adaboost-based model for security level classification of mobile intelligent terminals

the training set T_N to train the Adaboost algorithm model. Firstly, T_N is divided into two categories, where $y = 1, y = 2$ is a class and $y = 3, y = 4$ is another class. Then $T_N^{(1)}$ with new classification features is obtained as follows:

$$T_N^{(1)} = \{(X_1, y_1^{(1)}), (X_2, y_2^{(1)}), \dots, (X_N, y_N^{(1)})\} \tag{2}$$

where the classification features are:

$$y_i^{(1)} = \begin{cases} 1, \ y \in \{1, \ 2\}; \\ 0, \ \ y \in \{3, 4\}; \end{cases}, \quad i = 1, 2, \dots, N. \tag{3}$$

After that, the weight distribution of the data in $T_N^{(1)}$ is initialized, and the initial weight of each training sample is the same, that is, the initial weight distribution of the training dataset $D_1(i)$ is:

$$D_1(i) = (w_1, w_2, \dots w_N) = \left(\frac{1}{N}, \dots, \frac{1}{N}\right). \tag{4}$$

Then, we use t $(t = 1, 2, \dots)$ times iterative process to train the classifiers. The detailed steps are as follows:

Step 1 Using the training set with weight distribution D_t to obtain the t weak classifier

$$H_t(x) : X \to \{1, 0\} \tag{5}$$

Step 2 Calculate the classification error rate of the current weak classifier:

$$e_t = P(H_t(X_i) \neq y_i) = \sum_{i=1}^{N} w_{ti} I(H_t(X_i) \neq y_i) \tag{6}$$

Step 3 Calculate the weight coefficient of this weak classifier in the final classifier:

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 - e_t}{e_t}\right). \tag{7}$$

And the t weak classifier is trained

$$f_t(x) = \alpha_t H_t(x) \tag{8}$$

Step 4 Update the weight distribution of the training set T:

$$D_{t+1} = \frac{D_t(i)\exp(-\alpha_t y_i H_t(x_i))}{Z_t} \tag{9}$$

where Z_t is the normalization constant

$$Z_t = 2\sqrt{e_t(1-e_t)}. \tag{10}$$

At this point, combining each weak classifier:

$$f(x) = \sum_{t=1}^{T} \alpha_t H_t(x) \tag{11}$$

Now is the inspection step, if the classification error rate of the dataset reaches 0, ending the iteration, and the final strong classifier is:

$$H_{\text{final}} = \text{sign}(f(x)) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t H_t(x)\right). \tag{12}$$

Otherwise, enter the $t+1$ iteration until the classification error rate reaches 0.

After the first strong classifier is generated, we make all the data in the training set pass through the strong classifier and pick out the data with $H_{\text{final}}(X_i) > 0$, which represent the data classified into Levels 1 and 2. The remaining data are Levels 3 and 4. Analogously, we will classify the dataset of Levels 1 and 2. The classification feature for Level 1 is labeled as 1, and the classification feature for Level 2 is labeled as 0. The dataset of Levels 3 and 4 is processed similarly. Repeat the training process and the second and third layer strong classifiers will get:

$$\begin{aligned} H^{(2)}_{\text{final}} &= \text{sign}(f^{(2)}(x)) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t H^{(2)}_t(x)\right) \\ H^{(3)}_{\text{final}} &= \text{sign}(f^{(3)}(x)) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t H^{(3)}_t(x)\right). \end{aligned} \tag{13}$$

After generating, using the testing set T_P to evaluate the obtained three-layer strong classifier and verify the accuracy. If the accuracy δ does not meet the requirements, the training process is restarted until obtaining the final three-layer strong classifier with high enough accuracy. Finally, the Adaboost-based model for security level classification of mobile intelligent terminals is built completely. The complete data flow is shown in Fig. 6. According to this method, process of classifying the security level is shown in Fig. 7.

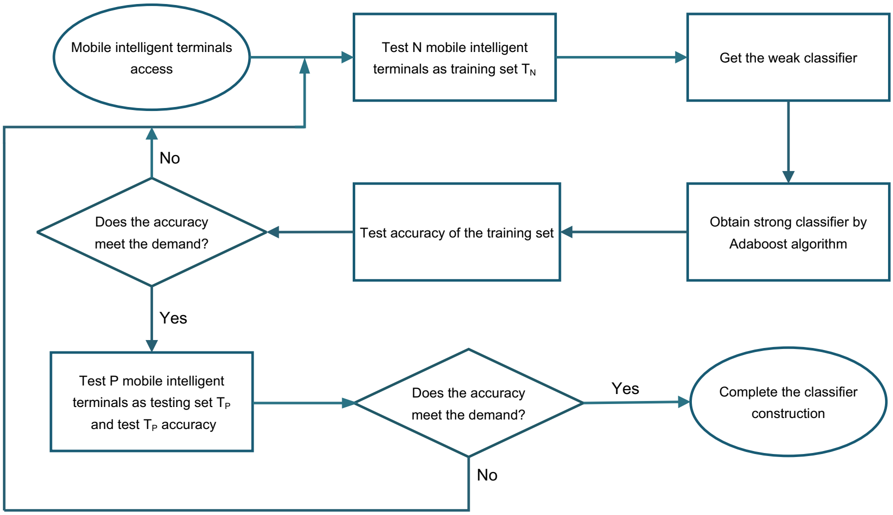

Mobile intelligent terminals access
Test N mobile intelligent terminals as training set T_N
Get the weak classifier
Obtain strong classifier by Adaboost algorithm
Test accuracy of the training set
Does the accuracy meet the demand?
No
Yes
Test P mobile intelligent terminals as testing set T_P and test T_P accuracy
Does the accuracy meet the demand?
Yes
No
Complete the classifier construction

Fig. 6 Processing flow of Adaboost mobile intelligent terminal security level model data

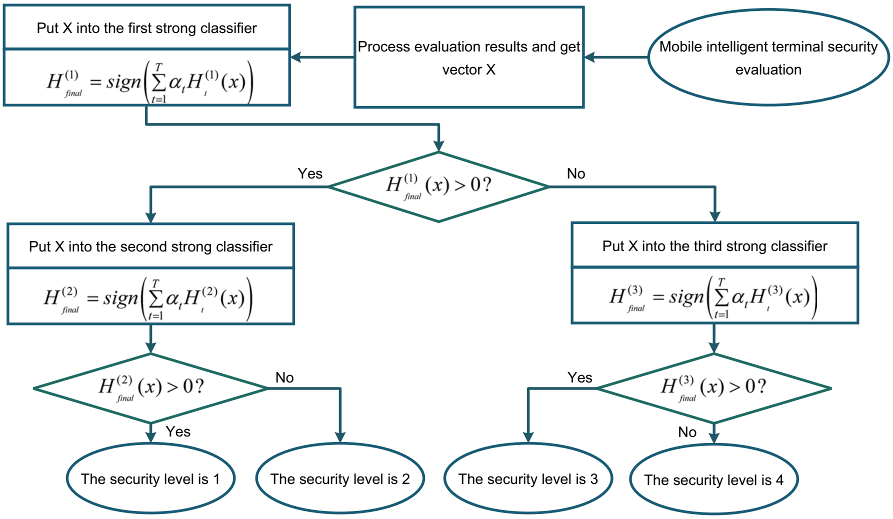

Put X into the first strong classifier
$H_{final}^{(1)} = sign\left(\sum_{t=1}^{T} \alpha_t H_t^{(1)}(x)\right)$
Process evaluation results and get vector X
Mobile intelligent terminal security evaluation
$H_{final}^{(1)}(x) > 0?$
Yes
No
Put X into the second strong classifier
$H_{final}^{(2)} = sign\left(\sum_{t=1}^{T} \alpha_t H_t^{(2)}(x)\right)$
Put X into the third strong classifier
$H_{final}^{(3)} = sign\left(\sum_{t=1}^{T} \alpha_t H_t^{(3)}(x)\right)$
$H_{final}^{(2)}(x) > 0?$
No
Yes
$H_{final}^{(3)}(x) > 0?$
Yes
No
The security level is 1
The security level is 2
The security level is 3
The security level is 4

Fig. 7 Process of Adaboost mobile intelligent terminal security level classification

Algorithm 1 Basic Method of Mobile Intelligent Terminal Security Classification

```
Input: matrix X of n test results; Counting rules Y for each item;
Input: the total number that each section contains: a,b,c,d
Output: Safety level vector: Level[N]
for i = 1 to $n$ do  // Go through each terminal.
    $Level[i] = 0$;
    if $X[i][1] = 1$ then  // Count the number of each stage.
        $A[1][1] = 1$; $A[1][2] = 0$; $A[1][3] = 1$; $A[1][4] = 0$; ← according to Y
    else
        $A[1][1] = 0$; $A[1][2] = 0$; $A[1][3] = 0$; $A[1][4] = 0$;
    end if
    ⋯
    if $X[i][22] = 1$ then
        $A[22][1] = 1$; $A[22][2] = 1$; $A[22][3] = 0$; $A[22][4] = 0$; ← according to Y
    else
        $A[22][1] = 0$; $A[22][2] = 0$; $A[22][3] = 0$; $A[22][4] = 0$;
    $a' = sum(A[\,][1])$; $b' = sum(A[\,][2])$;
    $c' = sum(A[\,][3])$; $d' = sum(A[\,][4])$;
    if $a'/a > 0.8$, Level[i]++; if $b'/b > 0.8$, Level[i]++;
    if $c'/c > 0.8$, Level[i]++; if $d'/d > 0.8$, Level[i]++;
    Level[i]++
end for
return Level[N]
```

3.3 Algorithms of two methods

This section expresses the algorithms of two methods above and calculates their time complexity. For the basic method, the detailed algorithm steps are shown in Algorithm 1. As shown in Algorithm 1, after the evaluation of N terminals, the security level of the terminal is obtained by statistics, proportion inspection and summation of each test result vector. For the second method, the detailed algorithm steps are shown in Algorithm 2. After the evaluation of N terminals, each testing result vector is judged by the three-layer strong classifier, and the security level of the terminal is obtained. For Algorithm 1, its time complexity is:

$$T_1(n) = 22n + 4n = 26n, \quad T_1(n) = O(n). \tag{14}$$

For Algorithm 2, its time complexity is:

$$T_2(n) = 3n, \quad T_2(n) = O(n). \tag{15}$$

It can be seen that although the two methods are at same level of time complexity, the first method takes much longer time than the second method.

Algorithm 2 Adaboost Method of Mobile Intelligent Terminal Security Classification

```
Input: matrix X of n test results
Input: strong classifier $H_1(x)$, $H_2(x)$, $H_3(x)$
Output: Safety level vector: Level[N]
for i = 1 to n do // Go through each terminal.
    Level[i] = 0 ;
    if sign($H_1(X[i] > 0)$ then // Import the Adaboost model.
        if sign($H_2(X[i] > 0)$ then
            Level[i] = 1; // Select the security level.
        else
            Level[i] = 2;
    else if sign($H_1(X[i] < 0)$ then
        if sign($H_3(X[i] > 0)$ then
            Level[i] = 3;
        else
            Level[i] = 4;
end for
return Level[N]
```

4 Simulation result and analysis

This section firstly writes a security assessment software to evaluate the security of the mobile intelligent terminal based on Sect. 2. Then, the Adaboost algorithm model is trained and tested by collected data and the error rate is detected. Finally, the time complexity of the basic method and the Adaboost method are compared for the practicability.

The dataset to use is the assessment results of 100 Android terminals that are obtained on the security assessment software from November 2017 to July 2018. Terminals include smartphones, smart bracelets, tablet PC and so on, from schools, hospitals, factories and other environments. After security evaluation, each terminal generates a set of 22-dimensional data, representing the test results of 22 test items. Among them, 60 pieces of data were randomly selected as a training set, and the remaining 40 pieces are testing set. In order to prevent overfitting, the data we use are screened to ensure diversity. Meanwhile, during the simulation, random noise is added to the expanded data to ensure the randomness. For each dataset, we perform 0, 1 numerical processing, so that each piece of data is a 22-dimensional row vector. Then, according to the security level determination method in Sect. 3, we calculate 100 pieces for security level. Among them, there are 0 data with security Level 1, 12 data with security Level 2, 37 with security Level 3 and 11 with security Level 4 for the training set. Besides, there are 0 data with security Level 1, 5 data with security Level 2, 30 data with security Level 3 and 5 data with security Level 4 for the testing set.

We implement both the basic method model and the Adaboost model in MATLAB 2017b and run it on the Window 7 platform of 3.2 GHz CPU and 8.00 GB RAM.

4.1 Mobile intelligent terminal security evaluation

Based on Sect. 2, we have to write a security assessment software to evaluate the security of the mobile intelligent terminal including location security, recording detection, browser recording security and other 22 test items. The evaluation result of each item is “No abnormalities” or “Disqualification.”

4.2 Simulation results

In this section, firstly the evaluation result is processed based on Sect. 3 to obtain 100 row vectors with 22 dimensions. Then, Fig. 8 shows the classification results of data. As we can see from Fig. 8a, the training set achieves complete classification correct by the strong classifier, and then the result of testing set from Fig. 8b, to

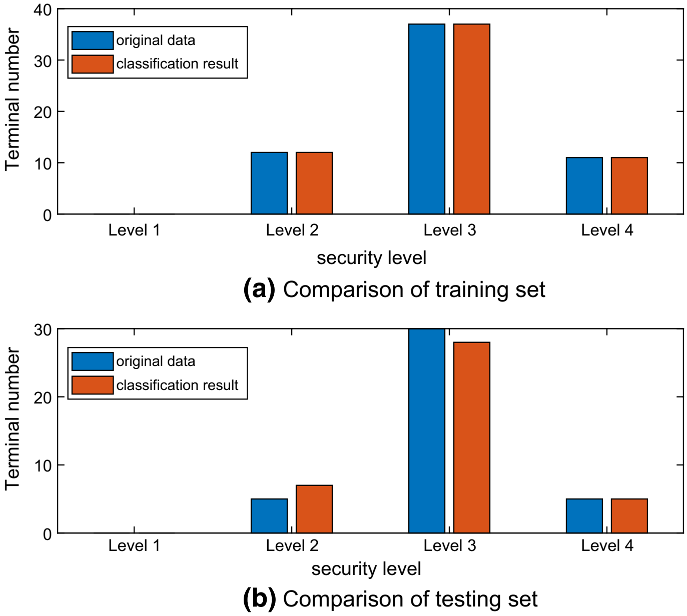


Fig. 8 Classification results of the training set and testing set on Adaboost

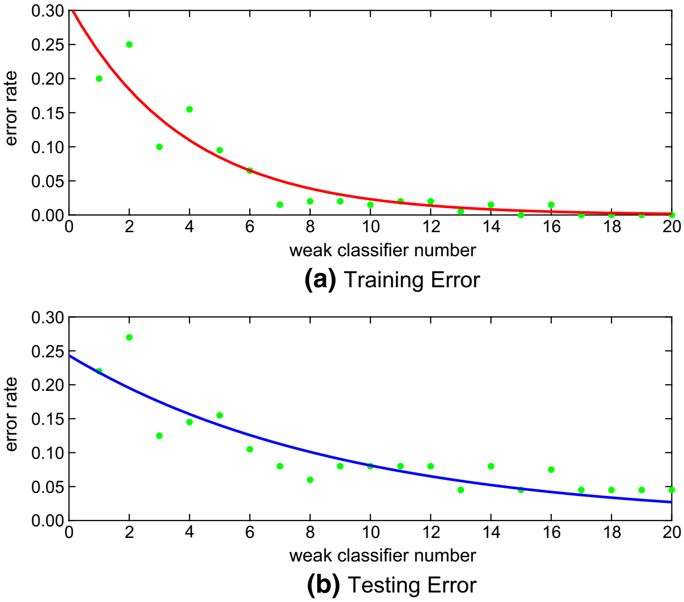


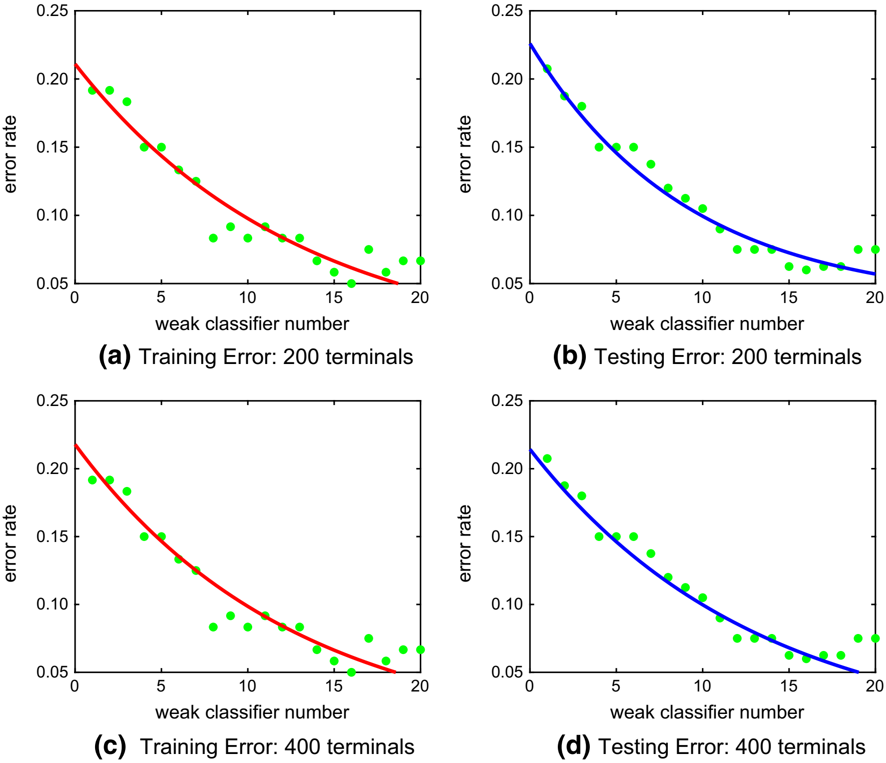
Fig. 9 Comparison of error rate between training set and test set

detect the correct rate of model, shows there are two wrong data classification and 38 correct data classification. The correct rate of Adaboost model is 95%.

Figure 9 shows the error rate movement of the training set and the testing set with the weak classifier number increases. In the model training process as Fig. 9a shows, the classification error rate decreases as the weak classifier number increases, and the classification error rate drops to 0 after the training of the 20th weak classifier. After that, we finish the strong classifier learning from training set. Meanwhile, we import the testing set to the model for comparison, and it can be seen from Fig. 9b that the classification error rate drops with the improvement in the classifier and finally falls to about 5%. The main reason of test error is from Levels 2 and 3 classification errors. This is because the training sample is small so that the feature learning of data is not complete enough. Although 5% error existing, the Adaboost algorithm has proved that it has a good learning effect.

4.3 Comparison of model accuracy rates under different data volumes

In order to verify the practicability of the above methods under larger data volume, we, respectively, simulate 200 and 400 security evaluation results of mobile intelligent terminals based on Monte Carlo algorithm and real dataset. We firstly carry out the security level definition according to the same process and then divide 200 data into 120 training set and 80 testing set, divide 400 data into 240 training set and 160 testing set. After that, we import them into Adaboost-based security



(a) Training Error: 200 terminals

(b) Testing Error: 200 terminals

(c) Training Error: 400 terminals

(d) Testing Error: 400 terminals

Fig. 10 Comparison of model error rates under 200 and 400 data volumes

level classification model to verify the accuracy of the model under different data amounts. The result is shown in Fig. 10.

We perform curve fitting for the classification error rate, and it can be seen that the classification error rate of the training set in different datasets shows a downward trend with the increase in weak classifiers. A small error rate is reached when the number of weak classifier reaches 20, and correspondingly the classification error rate of testing set also decreases. The final accuracy of both testing sets remained above 90%. Compared with the real data, the classification error rate does not change significantly as the number of data increases, and it remains at around 6%. Therefore, it can be considered that the proposed scheme still maintains a good classification when a large number of terminals are connected simultaneously. Figure 10 also shows that the error rates of both the training set and the testing set are the lowest when the number of weak classifiers reaches 16, which is due to the fact that choosing weak classifier is more flexible when the number of data increases, resulting in the probability increasing that the classification is completely correct, but the decreasing trend of the classification error rate is constant.

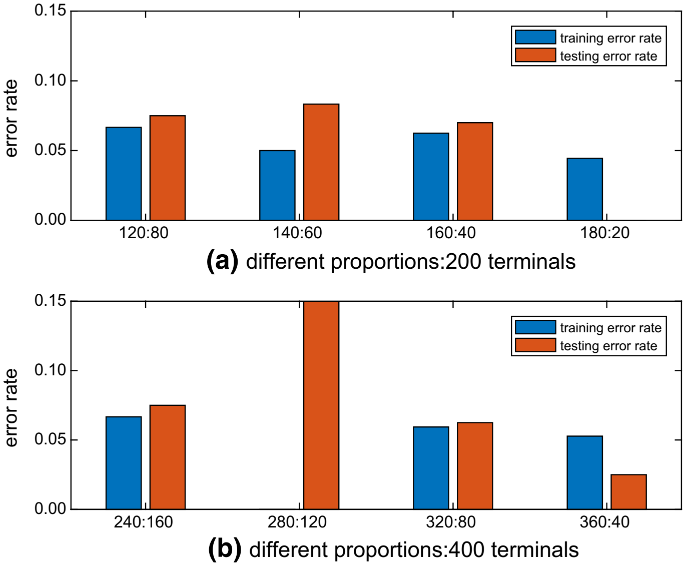


Fig. 11 Effect of different data divisions on the classification error rate

4.4 Influence on classification error rate under different proportion divisions

In order to discover a more reasonable proportional division of the training set and the testing set, we verify the effect of different data divisions on the classification error rate. The results are shown in Fig. 11.

It can be seen from Fig. 11a, b that in both 200 and 400 number of intelligent terminals, the classification error rate of testing set is the lowest when the ratio of training set to testing set is the largest. However, on this occasion, the amount of the testing data is too small to sufficiently express the classification accuracy of the model. The classification error rate of testing set is the highest when the rate of training set to testing set is 7:3, which indicates that the classification error of model gradually recovers with the data volume of the testing set increasing. The classification error rate is the best as the rate is 3:2, indicating the initial division rate has excellent rationality.

4.5 Time complexity analysis

Figure 12 compares the time complexity among NN, SVM and Adaboost in model training. We use the 400 dataset to train the NN and SVM model for security level classification. Here, we use a back-propagation neural network (BPNN) with two hidden discriminant layers. For visualized comparison, we set 20 as the number of support vectors of SVM algorithm, the number of nodes in each hidden discriminant layer of BPNN algorithm and the number of weak classifiers of Adaboost algorithm. It can be seen that as the number of accessed terminals increases, the

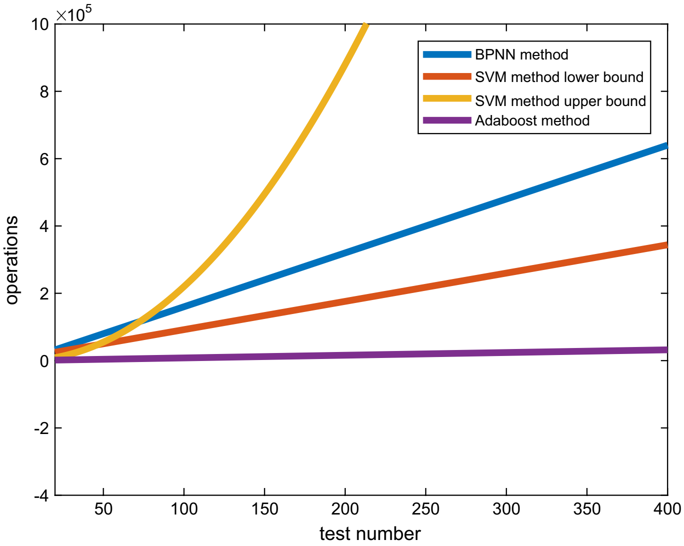


Fig. 12 Time complexity of Adaboost and similar algorithms

operation times of both BPNN algorithm and SVM algorithm are higher than that of Adaboost algorithm. Further, for accuracy improvement, the time complexity of SVM algorithm can be orders of magnitude higher than other methods. At the edge of the network discussed in this article, resource savings often exceed any accuracy improvement gained by complex algorithm design. We have proved above that

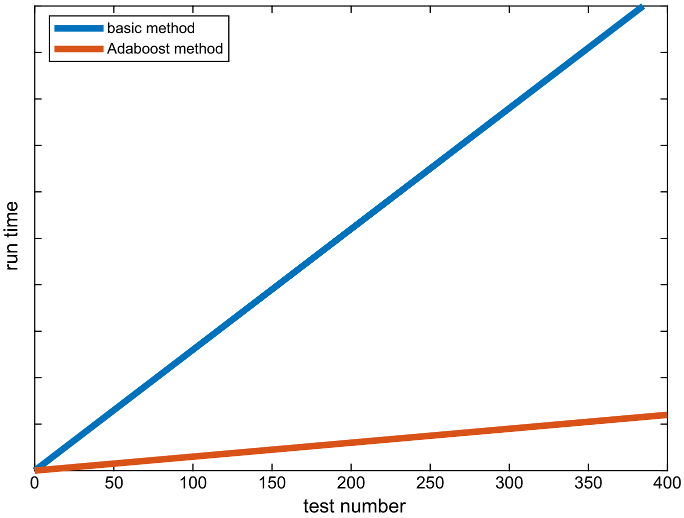


Fig. 13 Classification time of Adaboost and basic method

under this condition, the classification accuracy of Adaboost algorithm has reached 94%, realizing excellent security level classification. Therefore, the Adaboost algorithm can achieve an outstanding classification effect with smaller resource demand, which satisfies our definition of threshold in Sect. 1.

Figure 13 compares the classification time between the basic method and the Adaboost classification method under the same terminal number. It is obvious that the run time of Adaboost method is much lower than the basic method. It is proved that our method proposed not only does not require a large number of computing resources when model training, but also achieves a faster security level classification than the basic method when applying. Above all, the Adaboost-based security level classification method of mobile intelligent terminals achieves excellent classification result, assisting the edge server rapidly realizing the reasonable dispatch of mobile intelligent terminals.

5 Conclusions

The security level classification can help in realizing the reasonable use of mobile intelligent terminals. This paper discusses the security evaluation method of mobile intelligent terminals and proposes an Adaboost-based security level classification method, which, compared with the basic method, aims to satisfy low time complexity and high classification accuracy for the classification. Simulation results show that our method is promising.

Acknowledgements This work was supported by National Natural Science Foundation of China (No. 61571104), Sichuan Science and Technology Program (No. 2018JY0539), Key projects of the Sichuan Provincial Education Department (No. 18ZA0219), Fundamental Research Funds for the Central Universities (No. ZYGX2017KYQD170) and Innovation Funding (No. 2018510007000134). The authors wish to thank the reviewers for their helpful comments. Dr. Dingde Jiang and Dr. Hong Wen are corresponding authors of this paper.

References

1. Kim J, Jeon Y, Kim H (2016) The intelligent IoT common service platform architecture and service implementation. J Supercomput 74(9):4242–4260
2. Boubiche DE (2018) Advanced industrial wireless sensor networks and intelligent IoT. IEEE Commun Mag 56(2):14–15
3. Dai W, Qiu M, Qiu L et al (2017) Who moved my data? Privacy protection in smartphones. IEEE Commun Mag 55(1):20–25
4. Lee YK, Kim JN, Lim KS et al (2017) Secure mobile device structure for trust IoT. J Supercomput 2017(7):1–19
5. Islam N, Das S, Chen Y (2017) On-device mobile phone security exploits machine learning. IEEE Pervasive Comput 16(2):92–96
6. Kraijak S, Tuwanut P (2015) A survey on IoT architectures, protocols, applications, security, privacy, real-world implementation and future trends. In: Proceedings of the WiCOM'15, pp 1–6
7. Shi W, Cao J, Zhang Q et al (2016) Edge computing: vision and challenges. IEEE Internet Things J 3(5):637–646
8. Taleb T, Dutta S, Ksentini A et al (2017) Mobile edge computing potential in making cities smarter. IEEE Commun Mag 55(3):38–43

9. Wang J, Hong Z, Zhang Y et al (2018) Enabling security-enhanced attestation with intel SGX for remote terminal and IoT. IEEE Trans Comput Aided Des Integr Circuits Syst 37(1):88–96
10. Galdi C, Nappi M, Dugelay J et al (2018) Exploring new authentication protocols for sensitive data protection on smartphones. IEEE Commun Mag 56(1):136–142
11. Park KC, Shin DH (2017) Security assessment framework for IoT service. Telecommun Syst 64(1):193–209
12. Lee L, Chen C (2016) The power of Smartphones. Multimed Syst 21(1):87–101
13. Wang G, Song Q, Zhu X (2015) An improved data characterization method and its application in classification algorithm recommendation. Appl Intell 43(4):892–912
14. Jia Q, Guo L, Jin Z et al (2016) Privacy-preserving data classification and similarity evaluation for distributed systems. In: Proceedings of the ICDCS'16, pp 690–699
15. Cavalcanti K, Viana E, Lins F (2017) An integrated solution for the improvement of the mobile devices security based on the android platform. IEEE Lat Am Trans 15(11):2171–2176
16. Irwan D, Asnar Y, Hendradjaya B (2015) Confidentiality and privacy information security risk assessment for Android-based mobile devices. In: Proceedings of the ICoDSE'15, pp 1–6
17. Dong Z, Wu Y, Pei M et al (2015) Vehicle type classification using a semisupervised convolutional neural network. IEEE Trans Intell Transp Syst 16(4):2247–2256
18. Deng Y, Ren Z, Kong Y et al (2017) A hierarchical fused fuzzy deep neural network for data classification. IEEE Trans Fuzzy Syst 25(4):1006–1012
19. Guo Y, Jia X, Paull D (2018) Effective sequential classifier training for SVM-based multitemporal remote sensing image classification. IEEE Trans Image Process 27(6):3036–3048
20. Ding X, Jiang T, Zou W (2017) A new method of dynamic gesture recognition using Wi-Fi signals based on Adaboost. In: Proceedings of the ISCIT'17, pp 1–5
21. Jiang D, Wang W, Shi L et al (2018) A compressive sensing-based approach to end-to-end network traffic reconstruction. IEEE Trans Netw Sci Eng 5(3):1–12
22. Jiang D, Huo L, Song H (2018) Rethinking behaviors and activities of base stations in mobile cellular networks based on big data analysis. IEEE Trans Netw Sci Eng 1(1):1–12
23. Jiang D, Huo L, Li Y (2018) Fine-granularity inference and estimations to network traffic for SDN. PLoS ONE 13(5):1–23
24. Jiang D, Huo L, Lv Z et al (2018) A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. IEEE Trans Intell Transp Syst 19(10):3305–3319

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.