Hindawi
Security and Communication Networks
Volume 2023, Article ID 9870209, 1 page
https://doi.org/10.1155/2023/9870209

Retraction

Retracted: Research on Smartphone Trojan Detection Based on the Wireless Sensor Network

Security and Communication Networks

Received 3 October 2023; Accepted 3 October 2023; Published 4 October 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] G. Li, "Research on Smartphone Trojan Detection Based on the Wireless Sensor Network," *Security and Communication Networks*, vol. 2022, Article ID 2455102, 7 pages, 2022.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 2455102, 7 pages
https://doi.org/10.1155/2022/2455102

Research Article

Research on Smartphone Trojan Detection Based on the Wireless Sensor Network

Gang Li

Guangxi Police College, Nanning, Guangxi 530028, China

Correspondence should be addressed to Gang Li; 202007000047@hceb.edu.cn

Received 2 June 2022; Revised 18 June 2022; Accepted 27 June 2022; Published 9 July 2022

Academic Editor: C. Venkatesan

In order to reduce false positives of Trojan horse detection algorithms in smartphones, a voting algorithm based on multiple machine learning algorithms in wireless sensor networks was proposed. Through setting up the experiment, the preliminary preparations of the experiment, including sample set selection, feature set extraction method, and algorithm effect evaluation criteria, were described first. The K-nearest neighbors algorithm, random forests algorithm, support vector machine, and voting algorithm were compared. The experimental results showed that SVM and KNN algorithms took the shortest time, about 0.3 seconds. Judging by the test results, the voting algorithm still performed the best among the four algorithms as the voting algorithm was an extension of the three machine learning algorithms. In these randomly selected samples, malicious programs and nonmalicious programs were successfully distinguished by the voting algorithm. As the amount of test data increased, the test results would be closer to the actual situation. Namely, the voting algorithm would also have a small probability of false positives, which could meet the design requirements of the system. It was concluded that the method could effectively reduce false positives of the Trojan horse detection algorithm in smartphones.

1. Introduction

Wireless sensor networks (WSNs) are composed of micro-smart sensor nodes deployed in the monitoring area and connected to each other through wireless communication. Its purpose is to realize data perception, processing, control, and other functions. It is a special Ad-Hoc network and an important part of Internet of Things technology [1]. Wireless sensor network integrates wireless communication technology, sensor technology, Ad-Hoc network technology, embedded system technology, and distributed computing technology, which can be deployed in some complex and harsh areas for a long time to provide real-time monitoring services for user terminals, as shown in Figure 1. The sensor nodes cooperate with each other in the way of short-distance communication. After data collection, real-time transmission and processing are completed, and the wireless routing protocol is used to transmit the data to the base station or the server, so as to realize the interconnection between the physical world and the information world. WSNs has been widely used in battlefield reconnaissance, environmental acquisition, agricultural and industrial measurement and control, electronic medical health and smart home, and other fields due to its advantages of convenient deployment, strong reliability, self-composition network, and low cost. Therefore, WSNs has become the research focus of many researchers in the field of information technology. With the continuous popularization of AI technology, the application of sensor technology is also developing and the performance requirements of WSNs are also increasing [2].

2. Literature Review

In the process of network communication, it is difficult to protect information by using encryption and identity authentication alone. Therefore, it is indispensable to deploy defense systems on important nodes of network systems. Compared with a firewall, the intrusion detection system is an active defense technology, which complements some shortcomings of firewall technology. The intrusion detection

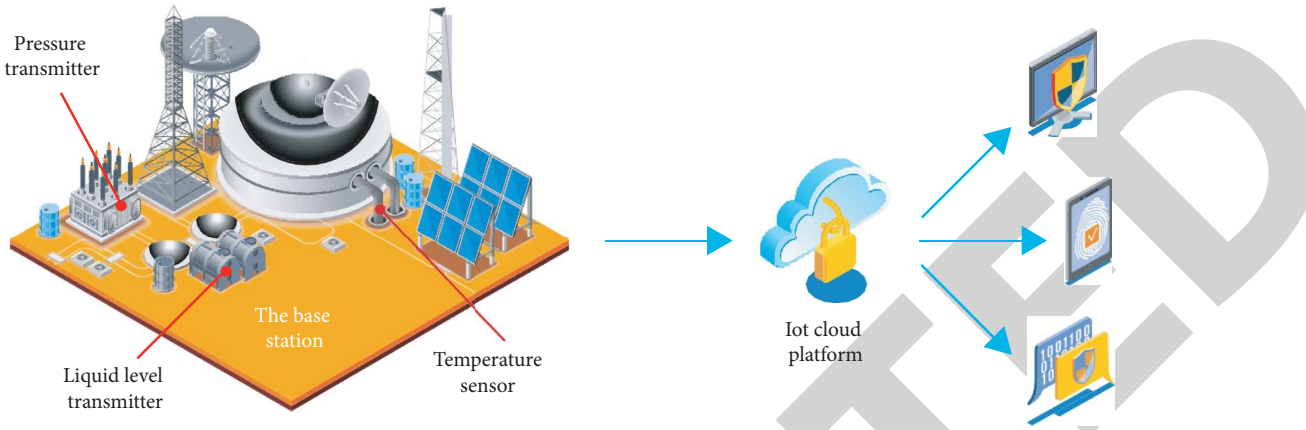

Figure 1: Wireless sensor networks.

system can proactively collect the information in the network or the system in real time, and then analyze and judge the information. If the corresponding security rules are violated, it will alarm and display the intrusion behavior, and finally, achieve the purpose of defense [3]. With the attention paid to wireless sensor network security at home and abroad, there have been many achievements in the research of wireless sensor network intrusion detection technology, the following are some representative research results.

Gavel et al. proposed a multi-agent distributed intrusion detection system applicable to clustered wireless sensor networks, which was composed of four agent modules, monitoring agent, detection agent, corresponding agent, and management agent, respectively, realizing the functions of data collection, detection analysis, intrusion response, and system management. The management agent was responsible for the management of the other three modules. These four agent modules were run on each sensor node. Each agent module could run independently and work together. The multi-agent distributed intrusion detection system could improve the scalability of the system, which was flexible and easy to implement in programming [4]. Zhao et al. proposed an intrusion detection model based on a repeated game with the penalty mechanism, mainly to constrain the existing attack nodes in wireless sensor networks and restrain their attack behaviors to reduce the harm to the network. The main idea was to analyze the repeated game process between nodes in the model and the intrusion detection system, to detect these attack nodes, and then to punish these attack nodes [5]. Mitra et al. proposed an intrusion detection model of wireless sensor networks based on the support vector machine, which was based on the network structure of clusters. The network was divided into three layers and each layer adaptively detected intrusion [6]. In order to reduce false positives of Trojan horse detection algorithms in smartphones, a voting algorithm based on multiple machine learning algorithms in wireless sensor networks was proposed. Through setting experiments, K-nearest neighbors algorithm, random forests algorithm, support vector machine, and voting algorithm were compared. Through the comparison of experimental results, it was proved that this method could effectively reduce the false positive problem of the Trojan detection algorithm in smartphones.

3. Research Methods

3.1. Heuristic Detection. The static and dynamic features of APK are extracted to form feature vector sets. On the one hand, the features of the training set samples can be extracted as the training data of the classifier. On the other hand, the APK features to be detected can be used as the data to be detected for classifier detection. Static features include permission feature, API call feature, and function feature file information, which can form multidimensional feature sets. Dynamic features include a series of action features generated during software triggering and execution, which can be extracted through log analysis [7]. The permission and API call features of software to be checked are extracted and identified by a classifier to determine whether it is a Trojan program. Android application permissions are declared in the Android manifest.xml file in advance. Undeclared permissions cannot be used for operating system data, such as connecting to the network and reading system status. During the installation, the application prompts the user with permission to install software, such as allowing the user to read SMS messages, send SMS messages, and read contacts. If the user does not agree, the application stops the installation. Permissions for applications can be extracted from the Android manifest.xml file [8]. When a Trojan horse steals user data to implement its own functions, it needs to call API. However, the API calls of a Trojan horse are different from those of normal programs. The class.dex file in the Android application APK decompression package saves the logical information of application software. The baksmali tool can be used to convert it into a smali code, from which API information can be extracted. The extracted feature information is trained by the machine learning algorithm and the Trojan horse can be detected by the classifier.

3.2. Feature Extraction. The feature extraction module is shown in Figure 2.

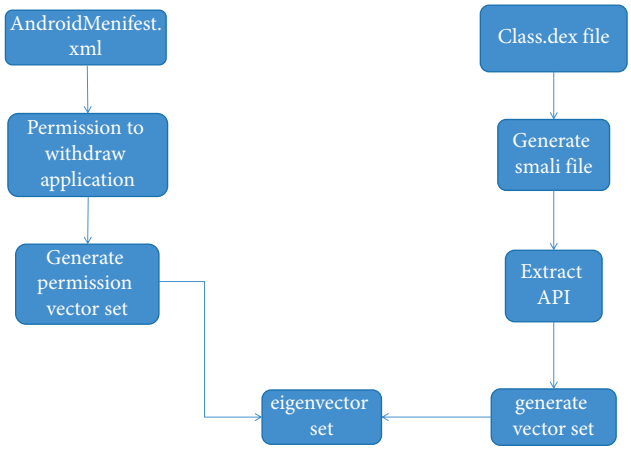

Figure 2: Feature extraction module.

Sensitive permissions such as WRITE_SMS are involved, which obviously exceeds the permissions required by normal software [9]. The applied permission in this file is compared with the sensitive permission list, "1" if applied and "0" if not applied, so as to form a feature vector of permission application. Android permission rules stipulate that applications must apply for permission in advance if they want to call sensitive API to prevent abuse of sensitive API by developers and protect the legitimate rights and interests of users. Sensitive API refers to some methods in the category that may affect user privacy. For example, the Trojan horse will call the API of sending SMS messages to the background. The baksmali tool is used to decompile the class.dex file in APK package to generate the smali file. The syntax of this file is very fixed and standard, from which API call information can be extracted. Sensitive API called from the smali file is extracted and compared with the sensitive API library. If API is called, it is marked as "1." And if it is not called, it is marked as "0," so as to form the feature vector of the calling of sensitive API [10].

3.3. Classifier Training. Machine learning algorithm is used to process the feature vectors extracted in the previous section and train the classifier. Trojan APK accumulated in the early case handling project and APK downloaded from the Internet are detected as normal software. Features are extracted to form the training sample set by the method in the previous section, and the machine learning algorithm in the next section is used to complete the training of the classifier. The classifier obtained can be used for the detection of unknown Trojans [11].

3.4. Machine Learning Algorithm. Machine learning is a form of computer work that relies on data rather than instructions. It is a method of generating models from existing data or experience and using that model to predict the future. The main problems to be solved in machine learning include classification, regression, and clustering [12]. There are many specific methods. In the application of Trojan detection, the commonly used methods are the following:

3.4.1. K-Nearest Neighbors Algorithm. K-nearest neighbors (KNN) algorithm is a basic machine learning method. The KNN method can perform both classification and regression. The three elements of the KNN algorithm are the selection of K value, distance measure, and classification decision rule. The basic idea of KNN in the Trojan horse detection algorithm is as follows: a training application set composed of n-dimensional feature vectors is given, for the applications to be detected in the test set, k applications closest to the application are found in the training application set. Most of the k applications belong to a malicious behavior category (or nonmalicious behavior), and the application is divided into the category [13].

KNN-based malicious behavior detection steps are as follows:

(1) Generating training sample set. The training sample application set is mapped to the corresponding feature behavior matrix according to its features.

(2) Selecting k recent samples. K samples with the highest similarity are selected from the training sample set.

(3) According to the classification to which the k nearest neighbors samples belong, the classification of test samples is decided.

When selecting the k value, a smaller value is generally selected according to the distribution of samples and an appropriate k value can be selected through cross verification. Choosing smaller values of k is equivalent to use smaller training instances in the field for prediction. Training error will decrease. Only training instances that the input instance is close to or similar with will work on forecast results. At the same time, the problem is that the generalization error will increase. In other words, the decrease of the k value means the whole model is complicated and overfitting is easy to occur [14]. Choosing a larger value of k is equivalent to using training examples in a larger field for prediction, which has the advantage of reducing generalization error, but it also has the disadvantage of increasing training error. At this time, training instances far from the input instance (dissimilar) will also act on the predictor, making the prediction produce error. And the increase of the k value means that the overall model becomes simple. One extreme is that k is equal to the number of samples m, so there is no classification at all. At this point, no matter what the input instance is, it simply predicts the category with the largest number of training instances it belongs to, and the model is too simple [15].

3.4.2. Random Forests Algorithm. Random forests algorithm is a kind of an integrated learning algorithm, which has the advantage of convenient parallel training and is suitable for data training of big data samples. In the training set of m samples, random samples are divided into T samples for training. Combined with the learning results of the weak learner, the strong learner is produced [16]. Random forests use the CART decision tree as a weak learner, and random forests improve the establishment of the decision tree.

Sample features of randomly selected nodes are assumed to be m. Among these randomly selected m sample features, the optimal sample is selected for division of the decision tree [17].

Trojan detection steps based on random forests are as follows:

(1) The sample set of the Trojan horse sample library is input and sample cases are selected by Bootstrap.

(2) On the basis of the features in the feature space, the decision tree is generated in each round. The important features are selected randomly to form a new feature set of Trojan horse detection. This new Trojan malicious behavior feature database is used to generate a decision tree [18].

(3) The final classification results are determined by voting. And the detection rate and the false positive rate are used to evaluate the detection ability.

3.4.3. Support Vector Machines. Support vector machines is a dichotomous model whose purpose is to find a hyperplane to segment the sample. The principle of segmentation is to maximize the interval, which is ultimately transformed into a convex quadratic programming problem to be solved [19]. The model from simplicity to complexity includes as follows: when the training sample is linearly separable, a linearly separable support vector machine is learned through hard interval maximization. When the training sample is approximately linearly separable, a linear support vector machine is learned by soft interval maximization. When the training sample is linearly untime-sharing, a nonlinear support vector machine is learned by the kernel technique and soft interval maximization.

Given training sample set $(D = (x_1, y_1)\ ,\ (x_2, y_2),\ \square, (x_m, y_m))$ $(D = (x_1, y_1)\ ,\ (x_2, y_2)\ ,\ )\ \square, (x_m, y_m)$, in which $y_i \in \{-1, +1\} y_i \in \{-1, +1\}$. The most basic idea of classification learning is to find a partition hyperplane in the sample space based on training set D. The important feature of support vector machines is that when training is completed, most samples do not need to be retained, and the final model is only related to support vectors.

The steps of SVM-based Trojan horse detection are as follows:

(1) To regularize the features extracted from the training sample base and construct the feature matrix.

(2) The Trojan horse samples and normal programs in the malicious behavior training sample library are divided according to their different characteristics by using the hyperplane of the SVM algorithm. The normal programs and Trojan horses are divided into different regions by the hyperplane.

(3) The test sample database is divided into the trained SVM hyperplane and the distribution of classification results is recorded. The detection ability is evaluated by the detection rate, false positive rate, missing negative rate, and other indicators.

3.4.4. Voting Algorithm Based on Various Machine Learning Algorithms of Wireless Sensor Networks. The machine learning algorithm classifier is used to determine the Trojan horse sample. If it is a Trojan horse, the algorithm outputs the result 1; otherwise, it outputs 0. With the increase of an excellent training set, the classification effect of the machine learning algorithm is getting better and better, but there still exist the problem of a high false positive rate which is difficult to be effectively eliminated. In order to improve the Trojan horse detection effect of the system, voting is designed for the output results of the machine learning algorithm. In this way, when at least two classifiers are determined to be Trojan horses, the module will determine them as Trojan horses to reduce the false positive rate of the system [20].

4. Result Analysis

4.1. Preparation for Experiment

4.1.1. Sample Preparation. The system designed in the research needed to be trained by the machine learning algorithm using normal software and Trojan software extracted from telecom fraud cases. Trojan horses found in the case were generally archived by the network security department and involved confidentiality. Because the test was to verify the effect of the algorithm, the Trojan horse sample set was replaced by the malicious program samples collected from the Virus Share website. Normal software was downloaded from Huawei application market, which had strict software review in China. It was guaranteed that the downloaded samples were nonmalicious normal programs. The categories covered shopping price comparison, audio and video entertainment, tools, etc., and the number of categories was about evenly distributed [21].

4.1.2. Feature Extraction. In order to simplify the test process and focus on the test of the algorithm detection effect, the experiment was simply characterized by the declaration of permissions in the extracted APK file. After cleaning the collected sample sets, 371 training set samples and 20 test set samples were used. There were 300 samples of the normal program training set and 20 samples of the test set. There were 135 permissions commonly used in the Android system, which were stored in the TXT file as a comparison table in advance. In order to reduce the workload and improve the efficiency of the experiment, a feature extraction program was designed using a Python language. For the programming implementation, the tool 7-ZIP was used to decompress the APK package after using the axmlPrinter2.jar tool to decompile the compiled Androidmanifest.xml and output the plaintext, and then the use-permission tag was matched to apply for the permission. By comparison with the permissions in the comparison table, if a certain permission was used, it output "1" in the corresponding position; otherwise, it output "0". And the direction table of APK system permission application was generated eventually [22].

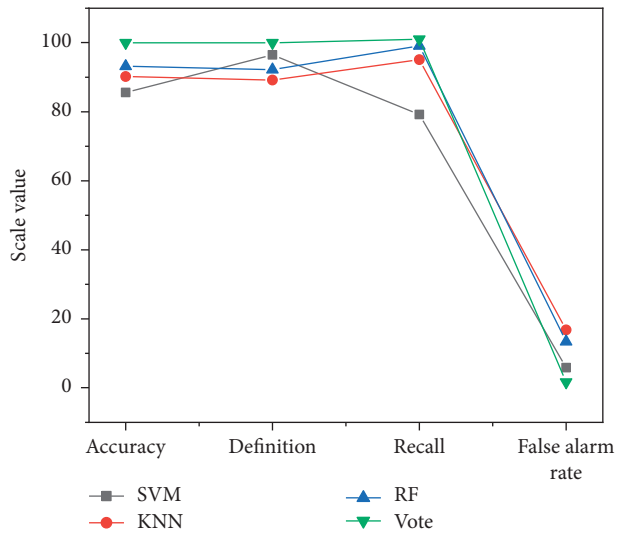

Figure 3: Test effect of the training set.

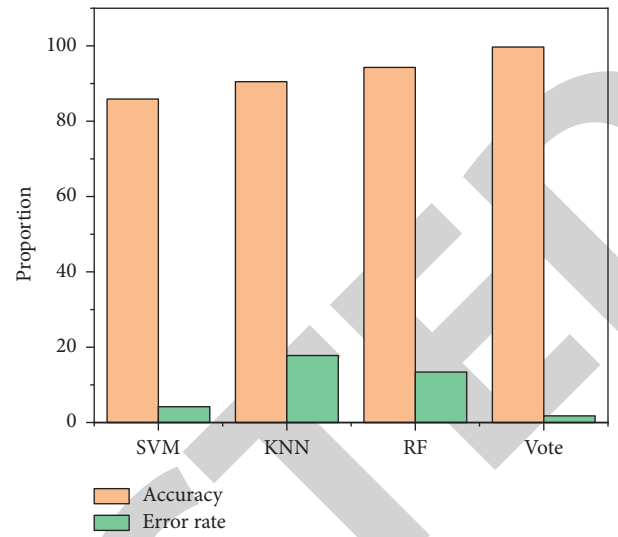

Figure 4: The accuracy and false positive rate of the training set algorithm.

4.1.3. Definition of Evaluation Criteria. The parameters used to evaluate the machine learning effect include TP (true positive), FP (false positive), FN (false negative), and TN (true negative). The corresponding indicators are as follows:

$$\begin{aligned}
\text{Accurate rate}: \text{TPR} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\
\text{False positive rate}: \text{TNR} &= \frac{\text{FP}}{\text{FP} + \text{TN}}, \\
\text{The recall rate}: \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\
\text{Accuracy}: \text{Accurary} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}.
\end{aligned} \tag{1}$$

4.1.4. Experimental Design. In order to verify the effect of several machine algorithms, the training sample set was used to train the classifier with three machine learning algorithms, namely KNN, FR, and SVM. And the output of the three classifiers was fused with voting rules as the fourth classifier. After the classifier training was completed, the training set and the test set were input to verify the effect of the algorithm. The whole experiment was realized by MATLAB [23].

4.2. Experimental Results and Analysis. After the classifier was trained with the training set, the classifier effect was tested with the training set data, as shown in Figure 3 for each evaluation index data.

The specific accuracy and the false positive rate are shown in Figure 4.

It can be seen that the training effect of the voting algorithm is better than the other three machine algorithms. And the index data of accuracy and false positive rate have great advantages. The training time statistics, specific

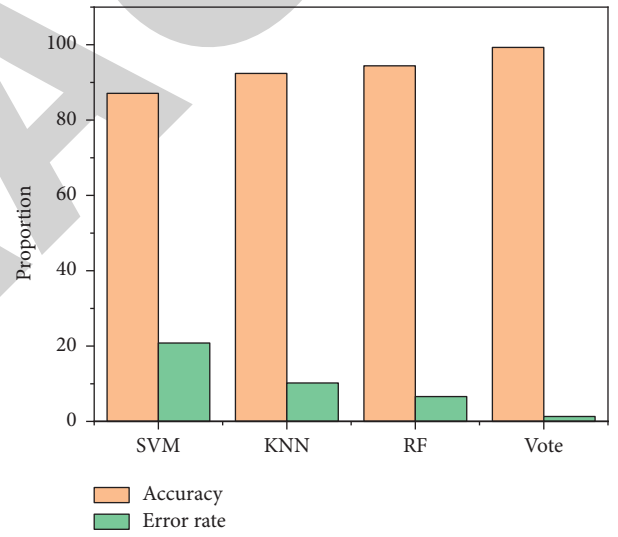

Figure 5: The accuracy and false positives rate of the testing set.

accuracy, and false positive rate results of the four algorithms are shown in Figure 5.

SVM and KNN algorithm took the shortest, about 0.3 seconds. Because it is based on three kinds of machine learning algorithms of the expansion, the training time of the voting algorithm is longer than the other three algorithms, which is expected. The consumption of time is in an acceptable range. It can improve accuracy obviously and reduce the rate of false positives significantly. However, in the application of the real system, the training of the algorithm can be carried out by multi-thread parallel calculation to improve the training efficiency of the voting algorithm [24]. The testing sample data is used to test the classifier, and the index data is shown in Figure 6.

It can be seen from the test results that the voting algorithm still performs best among the four algorithms. And in the randomly selected samples, malicious programs and

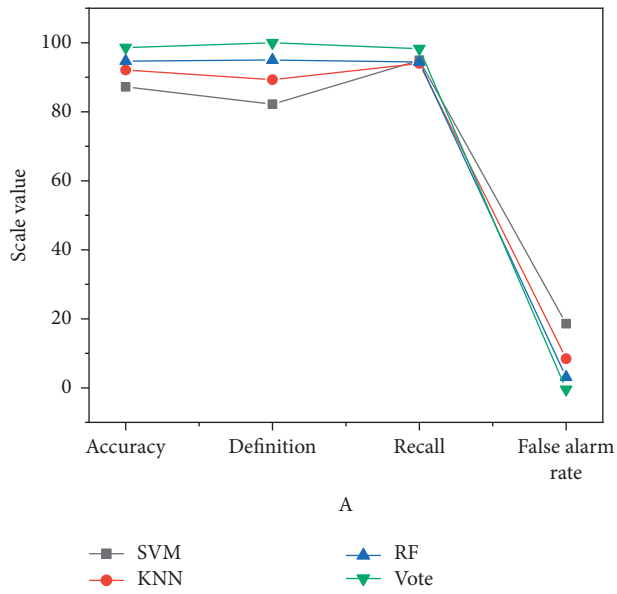

Figure 6: Test effects of the testing set.

nonmalicious programs is successfully distinguished by the voting algorithm. This is ideal for the real-world use due to the small amount of test data. As the amount of test data increases, the test results will be closer to the actual situation [25], that is, the voting algorithm will also have a small probability of false positives, which can meet the design requirements of the system.

5. Conclusion

In the research, MATLAB was used to verify the effectiveness of the algorithm. Firstly, the preliminary preparations of the experiment, including sample set selection, feature set extraction method, and algorithm effect evaluation criteria, were described. Subsequently, SVM, KNN, RF, and voting algorithm classifiers were, respectively, trained with the training set on MATLAB and tested with the training set and the sample set. Training accuracy and test accuracy as well as several related evaluation indicators were obtained. Experimental results showed that the voting algorithm based on SVM, KNN, and RF was better than the first three algorithms in both the accuracy and false positives rate.

In the research, some achievements have been made in these two aspects, but there are still some shortcomings to be improved.

(1) The Android Trojan horse detection system for public security business proposed in the research is still in the design stage and the technology used in the system is not described in detail due to the length. In the later stage, it is necessary to further supplement and improve the design scheme to form a specific scheme and develop it, so as to complete the project landing and serve the investigation of telecom fraud cases.

(2) The new algorithm voting rules are relatively simple. And the weight parameters of three machine learning algorithms need to be determined through more experiments in the later stage to further improve the accuracy of the algorithm.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The study was supported by the Project Research and Practice on Teaching Reform of Public Security Based on the Mobile Network Security Information Platform, special project of the 2019 Guangxi Higher Education Undergraduate Teaching Reform Foundation (No. 2019JGB402Li).

References

[1] S. H. Gopalan, "Zhrp-dcsei, a novel hybrid routing protocol for mobile ad-hoc networks to optimize energy using dynamic cuckoo search algorithm," *Wireless Personal Communications*, vol. 118, no. 4, pp. 3289–3301, 2021.

[2] Y. Zhang, C. Chen, Y. Zheng, Y. Shao, and C. Sun, "Application of fiber Bragg grating sensor technology to leak detection and monitoring in diaphragm wall joints: a field study," *Sensors*, vol. 21, no. 2, pp. 441–450, 2021.

[3] X. He, "Research on computer network security based on firewall technology," *Journal of Physics: Conference Series*, vol. 1744, no. 4, pp. 042037–024043, 2021.

[4] S. Gavel, A. S. Raghuvanshi, and S. Tiwari, "Distributed intrusion detection scheme using dual-axis dimensionality reduction for internet of things (iot)," *The Journal of Supercomputing*, vol. 77, no. 2, pp. 1–24, 2021.

[5] N. Zhao, Z. Sheng, and H. Yan, "Emission trading innovation mechanism based on blockchain," *Chinese Journal of Population, Resources and Environment*, vol. 19, no. 4, pp. 8–15, 2021.

[6] A. Mitra, G. Jana, R. Pal, P. Gaikwad, S. Sural, and P. K. Chattaraj, "Determination of stable structure of a cluster using convolutional neural network and particle swarm optimization," *Theoretical Chemistry Accounts*, vol. 140, no. 3, pp. 1–12, 2021.

[7] J. Zhaoxue, L. Tong, Z. Zhenguo, G. Jingguo, Y. Junling, and L. Liangxiong, "A survey on log research of aiops: methods and trends," *Mobile Networks and Applications*, vol. 26, no. 6, pp. 2353–2364, 2021.

[8] D. Kumar, A. Sharma, R. Kumar, and N. Sharma, "Restoration of the network for next generation (5g) optical communication network 2019," in *Proceedings of the International Conference on Signal Processing and Communication (ICSC)*, IEEE, Noida, India, March 2019.

[9] J. Huang, W. Huang, F. Miao, and Y. Xiong, "Detecting improper behaviors of stubbornly requesting permissions in android applications," *International Journal on Network Security*, vol. 22, no. 3, pp. 383–393, 2020.

[10] J. Tang, Y. Wang, C. Wang, H. Huang, N. Liu, and N. Al-Nabhan, "Image edge detection based on singular value feature vector and gradient operator," *Mathematical Biosciences and Engineering*, vol. 17, no. 4, pp. 3721–3735, 2020.

[11] M. Bradha, N. Balakrishnan, S. Suvi et al., "Experimental, computational analysis of butein and lanceoletin for natural dye-sensitized solar cells and stabilizing efficiency by iot," *Environment, Development and Sustainability*, vol. 24, 2021.

[12] Z. Huang, "Accurate recognition method of continuous sports action based on deep learning algorithm," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 3407935, 10 pages, 2022.

[13] M. Castellanos González, B. Díaz-Ley, and M. A. Segurado Rodríguez, "Cuando tomarse un gin-tonic se convierte en una mala experiencia: exantema fijo medicamentoso por quinina," *Actas Dermo-Sifiliográficas*, vol. 111, no. 2, pp. 178–180, 2020.

[14] A. Malikov, "Adaptation of the diagnostic artificial neural network structure when new training examples appear," *Proceedings of Telecommunication Universities*, vol. 6, no. 4, pp. 120–126, 2020.

[15] Z. Li, Y. Fan, and L. Ying, "Multilevel fine-tuning: closing generalization gaps in approximation of solution maps under a limited budget for training data," *Multiscale Modeling and Simulation*, vol. 19, no. 1, pp. 344–373, 2021.

[16] X. Liu, J. Liu, J. Chen, and F. Zhong, "Catalytic conversion and DFT analysis of the post DDBD-catalysis system for degradation of toluene, ethyl acetate and acetone with different metal-oxides catalysts," *Journal of Rare Earths*, vol. 13, 2022.

[17] D. R. Dadsena, Sadukhan, and R. Swaminathan, "Rotational moment shape feature extraction and decision tree based discrimination of mild cognitive impairment conditions using mr image processing," *Biomedical Sciences Instrumentation*, vol. 57, no. 2, pp. 228–233, 2021.

[18] R. Huang, S. Zhang, W. Zhang, and X. Yang, "Progress of zinc oxide-based nanocomposites in the textile industry," *IET Collaborative Intelligent Manufacturing*, vol. 3, no. 3, pp. 281–289, 2021.

[19] N. N. Tyunin, "Problems of nonconvex quadratic programming related to the optimization of phased antenna arrays," *Journal of Applied and Industrial Mathematics*, vol. 15, no. 3, pp. 543–557, 2022.

[20] B. Solihah, A. Azhari, and A. Musdholifah, "The empirical comparison of machine learning algorithm for the class imbalanced problem in conformational epitope prediction," *JUITA: Jurnal Informatika*, vol. 9, no. 1, pp. 131–147, 2021.

[21] J. James, A. R. Annamalai, A. Muthuchamy, and C. P. Jen, "Effect of wettability and uniform distribution of reinforcement particle on mechanical property (tensile) in aluminum metal matrix composite-A review," *Nanomaterials*, vol. 11, no. 9, pp. 2230–2236, 2021.

[22] M. Jafari, K. Abbaszadeh, and M. Mohammadian, "A 12-sector space vector switching table for parallel-connecting to dual induction motors fed by matrix convertor based on direct torque control," *SN Applied Sciences*, vol. 3, no. 11, pp. 1–14, 2021.

[23] Y. Zhang, X. Kou, Z. Song, Y. Fan, M. Usman, and V. Jagota, "Research on logistics management layout optimization and real-time application based on nonlinear programming," *Nonlinear Engineering*, vol. 10, no. 1, pp. 526–534, 2021.

[24] X. Zhang, Z. Z. Zhang, R. L. Xian, H. P. Fang, and Y. Hu, "Status survey and training efficiency of food allergy knowledge among pediatric medical worker," *Zhonghua er ke za zhi. Chinese journal of pediatrics*, vol. 58, no. 9, pp. 753–757, 2020.

[25] W. Shen, J. Sun, F. Yao et al., "Microbiome in intestinal lavage fluid may be a better indicator in evaluating the risk of developing colorectal cancer compared with fecal samples," *Translational Oncology*, vol. 13, no. 5, pp. 100772–100780, 2020.