Hindawi
Security and Communication Networks
Volume 2021, Article ID 9396141, 12 pages
https://doi.org/10.1155/2021/9396141

Research Article

Attribution Classification Method of APT Malware in IoT Using Machine Learning Techniques

Shudong Li,[1] **Qianqing Zhang**,[1] **Xiaobo Wu**,[2] **Weihong Han**,[1] **and Zhihong Tian**[1]

[1]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China*
[2]*School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China*

Correspondence should be addressed to Shudong Li; lishudong@gzhu.edu.cn

Received 10 June 2021; Accepted 23 August 2021; Published 7 September 2021

Academic Editor: Shui Yu

In recent years, the popularity of IoT (Internet of Things) applications and services has brought great convenience to people's lives, but ubiquitous IoT has also brought many security problems. Among them, advanced persistent threat (APT) is one of the most representative attacks, and its continuous outbreak has brought unprecedented security challenges for the large-scale deployment of the IoT. However, important research on analyzing the attribution of APT malware samples is still relatively few. Therefore, we propose a classification method for attribution organizations with APT malware in IoT using machine learning. It aims to mark the real attacking organization entities to better identify APT attack activity and protect the security of IoT. This method performs feature representation and feature selection based on APT behavior data obtained from devices in the Internet of Things and selects the features with a high degree of differentiation among organizations. Then, it trains a multiclass model named SMOTE-RF that can better deal with imbalance and multiclassification problems. Our experiments on real dynamic behavior data are combined to verify the effectiveness of the method proposed in this paper for attribution analysis of APT malware samples and achieve good performance. Our method could identify the organization behind complex APT attacks in IoT devices and services.

1. Introduction

As IoT applications and services spread to every corner of our lives, the number of Internet of Things devices is rapidly increasing, However, most of the devices were developed without considering security issues, as well as cannot be updated, which makes cybercriminals vulnerable to attack when they find some errors or security problems. Ubiquitous IoT has brought many security problems [1, 2]. The VPNFilter incident was one of the most serious IoT device attacks of 2018, and the US Department of Justice has since linked the incident to APT28. The incident affected 50W devices in at least 54 countries and regions worldwide, affecting and damaging the ubiquitous IoT environment. In August 2019, Microsoft reported that its Threat Intelligence Center had detected an attack on IoT devices, including VoIP phones, printers, and video decoders. Two of the three devices affected by the attack had factory security settings, while the software in the third was not updated. Microsoft blamed the attack on a Russia-based group commonly known as APT28. As AI advances, hackers are also using it to launch more sophisticated attacks on computer systems. Among the attacks of the IoT, advanced persistent threat (APT) is one of the most representative attacks, and its continuous outbreak has brought unprecedented security challenges. Therefore, the APT attack has attracted much attention of various researchers and many governments. An APT attack is a form of long-term and persistent network attack by individuals or organizations, who use advanced attack techniques against specific targets. The difference between the APT attack and the traditional network attack is that the APT attack has the characteristics of concealment, pertinence, persistence, and organization [3]. Its attack means are changeable, the attack effect is remarkable, and it is difficult to prevent, such as the famous APT attack, the "Stuxnet" virus [4]. The virus broke out in 2010. The technology of the virus is complex and hidden, which makes the discovery and analysis process take long time. Its infection was targeted mainly at Iran's nuclear facilities, which had a huge impact on Iran's nuclear program. This

incident is considered an organized state act. Moreover, in 2016, hackers launched DDOS attacks by manipulating IoT devices infected with malware known as Mirai. Behind APT attacks, there are usually organizations with government background or intelligence institutional background that provide funding with political or economic purpose [5]; the threat to national and enterprise information security systems is becoming more and more serious, and the number of APT reports is increasing year by year. Security agencies of various countries have disclosed hundreds of APT organizations, commonly active ones being Russia's APT28 and APT29, North Korea's Lazarus, and so on. Attribution analysis of APT samples has always been one of the most important links in the analysis of APT attacks, and it is also a method to detect APT attacks [6]. At present, industrial analysis on the attribution of APT samples mainly relies on the manual analysis by safety experts, which are greatly affected by the expert experience. Besides, it cannot meet the need of a large number of samples, which are low in efficiency and time-consuming. There are relatively few studies on the attribution of attack samples in academia. With the rapid increase in the number of polymorphic viruses and deformed Trojans, malware has become one of the usual methods of APT attacks [7]. FireEye is proposed to perform APT organization clustering based on the similarity of malicious code samples [8]. The characteristics of malware are mainly divided into static features (binary file characteristics, disassembly features, etc.) [9] and dynamic features (execution behavior features, etc.) [10]. Static features are generally disassembled, etc. It is usually difficult to extract effective features due to polymorphism, deformation, and shelling. Dynamic features are generally obtained by monitoring the behavior of the program during runtime, which is not affected by confusion technology [11–14].

APTs of the same organization have certain similarities in their behavior, to realize the automatic classification of APT malware samples, that is, to classify and identify the samples of the same organization. Based on the behavioral data of APT attack malware obtained from the Internet of Things devices, this paper proposes a classification method of APT attack organization based on machine learning. The main contributions of this paper are as follows:

(i) We propose an APT organization classification method based on machine learning and malware. The method that aims to effectively identify APT attack activity has been verified by experiments in that it has stable performance and high efficiency, which can mark real attack organization entities to protect the security of the Internet of Things.

(ii) We carry out feature representation and selection filtering in that to get the features with a higher distinguishing degree in different organizations based on the acquired behavior data of malware, which reduces the feature dimension and improves the calculation speed.

(iii) Due to the imbalance of the APT organization data set, we designed the SMOTE-RF model to solve this multiclassification problem.

1.1. Related Works. The APT attack is a complex network attack with a very obvious purpose. It attacks the target network step by step through multiple stages and maintains long-term access to the target [15]. With the aid of APT malware, an attacker can remotely control the infected machine and steal sensitive information. APT malware, such as Trojan horses or backdoors, is a firewall dedicated to antivirus software and target networks. It is not only used to remotely control the infected machine in APT attacks but also used to steal sensitive information from the infected host for a long period [16].

Currently, commonly used detection methods related to APT are mainly researched from the aspects of malicious code detection, attack detection, and network traffic detection. Abomhara and Kien [17] proposed threats and attacks faced by the IoT infrastructure. In addition to analyzing and describing intruders and attacks faced by IoT devices and services, they also tried to classify threat types. Sung et al. [18] proposed a new and practical security architecture model that protects each layer and interface. The protection includes data protection, access control, prevention of threats, and protection against network attacks. By mapping these analytical protection controls to the risks in each department and its resources, companies can apply robust multilayer defenses against any attack, including advanced persistent threats. Lee and Lewis [19] proved that it is possible to use undirected graphs to associate attacks based on shared targets between different attacks. Based on this information, a map of APT activities can be built and clusters that may represent the activities of a single team of malware writers can be identified. In addition, there are some other detection methods [20, 21].

For the detection of malicious software, malware can be identified by intelligent analysis of the characteristics of malicious samples [22]. As malware has become a necessary strategic tool for APT attacks, the characteristics of malware can also be used as the characteristics of APT attack organizations [23]. The methods of extracting malware features mainly include static feature extraction and dynamic feature extraction [24]. The static feature extraction method uses file structure analysis, decompilation, disassembly, control flow, and data flow analysis techniques to extract static features such as component instructions, control flow, and function call sequence of the program without running the program. For example, Qiao et al. [25] proposed an automatic malware homology identification method based on API calls. This method obtains its API set through static analysis of malicious samples, then uses the Jaccard similarity coefficient to calculate the homology degree of different malware types based on the six calling behaviors defined by

programming habits, establishes a threshold to compare with the homology degree through experience, and draws a conclusion about whether the samples are similar or not. This method can be used to determine the degree of homology between APT samples and determine the organization of the samples. The dynamic feature extraction is used to monitor the behavior of the program when it is running and then extract the dynamic behavior characteristics of the code such as API operations, file system operations, function access, and system calls. For example, Chen et al. [5] proposed a new genetic model combined with a knowledge map of malware behavior. Their method is to build a genetic model based on the content of the node, extract the gene sequence of all malware belonging to each APT organization, and then calculate the similarity between the malware and the gene library and compare it with a preset threshold like which APT organization of the malware belongs to.

In the industry, APT organizational identification is more inclined to analyze the correlation between malicious code structure and its attack chain. For example, FireEye Lab [26] analyzed 11 APT attacks in 2013 and found the malicious code used in the attack based on the same code segment, timestamp, digital certificate, etc. Based on these collected characteristics, the correlation analysis is carried out, and it is believed that the attacks are all manipulated by the same organization. Beijing Venustech Inc. [27] analyzes the shellcode function and code similarity of some samples of vulnerabilities as the characteristics of correlation analysis and then traces the source of the Hedwig organization. Lockheed Martin proposed an advanced continuous threat kill chain model [28], which divided APT attack activities into Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command and Control, and Actions on Objectives, They performed these 7 steps and pointed out that as long as the defending party detects and blocks one of these steps, the attack can be prevented from occurring. MITRE Company adopted a similar idea and proposed a more detailed ATT&CK framework [29]. ATT&CK integrates known historical and actual advanced threat attack tactics and technologies to form a common language for hacking descriptions and an abstract knowledge base framework for hacking attacks. Its official website has published descriptions of 87 APT attacking organizations, including the software, tactics, techniques, and procedures (TTP) used by the attacking organizations.

2. The Proposed Method

This paper proposes a classification method for attribution organizations with APT malware based on machine learning. Based on malicious software samples in APT attacks, this method first dynamically analyzes samples, preprocesses the acquired behavior data, constructs a behavioral data set of malware samples, then uses the TF-IDF method to perform the feature representation forms a vector matrix, and calculates the chi-square value of the high-latitude feature vector to perform feature selection. Based on the SMOTE-RF model designed in this paper, the multiclass model is trained and finally, the test set is predicted and output. The overall design framework of this article is shown in Figure 1.

2.1. Feature Representation. In this paper, we use the behavior feature data set that is the APT data set provided by NSFOCUS. They collected and obtained the dynamic information of a large amount of malware in the sandbox and marked the APT organization to which it belongs. This experiment selected sample data of 7 APT organizations to form the original data set, and the information is shown in Table 1.

The behavioral data of the samples in this dataset contain a lot of redundant data, including path data generated when the malware executes operations, various files called by the malware, APIs, operation object data, and other information (see Figure 2).

The behavior data of a malware sample is in the form of text, as shown in the diagram of a sample behavior data of a malware sample. Therefore, before model training, the text data must be quantified. According to our statistics, the text character length of most samples is below 10,000, so the first 10,000 characters are intercepted for the text data of each sample (see Figure 3). Then, we choose to use TF-IDF (term frequency-inverse document frequency), that is, the word frequency and inverse document frequency method to weight each word to vectorize the text. The TF-IDF method consists of two parts: term frequency (TF) and inverse document frequency (IDF). The former represents the frequency of a term in a document and is used to describe the importance of a term to a document. The latter represents the proportion of documents that contain a certain term and is used to measure whether the term is common or rare in all documents.

If the $n_{i,j}$ represents the frequency of the term v_i in the document d_j, the word frequency of the term v_i represents as

$$\mathrm{TF}_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,k}}. \tag{1}$$

Inverse text frequency of term $\mathrm{IDF}_{i,j}$ is expressed as

$$\mathrm{IDF}_{i,j} = \log \frac{|D|}{\left\| d_j : v_i \in d_j \right\| + 1}, \tag{2}$$

where $|D|$ represents the total number of all documents in the corpus and $|d_j : v_i \in d_j|$represents the total number of documents in the corpus containing the term v_i. Therefore, the term frequency and inverse text frequency $\mathrm{TF} - \mathrm{IDF}_{i,j}$ of the term v_i in the document d_j are expressed as

$$\mathrm{TF} - \mathrm{IDF}_{i,j} = \mathrm{TF}_{i,j} \times \mathrm{IDF}_{i,j}. \tag{3}$$

If the TF value of a word extracted from the behavior data is very high but the IDF value is very low, it indicates that the word may be important to the attack.

When using the TF-IDF algorithm to identify keywords in behavioral data, treat all the data extracted from the same sample as an independent document d_j, and all d_j constitute a corpus. Calculate the TF-IDF value for each word v_i in d_j. To save costs, improve efficiency, and reduce false alarms, a set of "stop words" is constructed as a white list. By default, these words are used very widely and do not affect classification. This set includes common English words in the

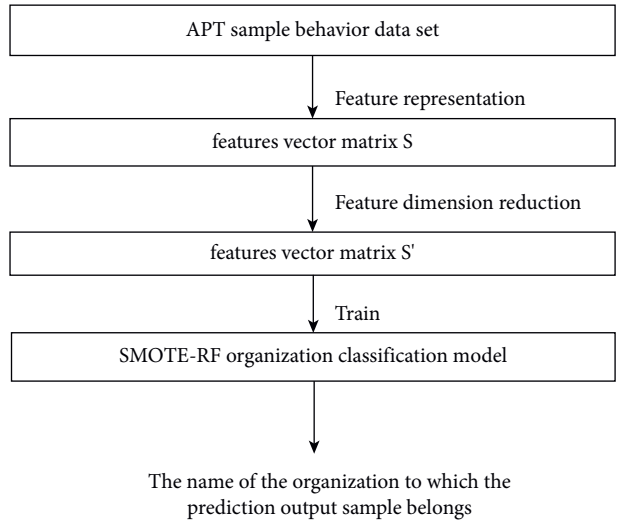

Figure 1: Overall design framework diagram.

Table 1: Distribution of the APT organization.

	APT organization name	Numbers
1	Lazarus	1060
2	APT28	343
3	Operation C-Major	276
4	APT29	273
5	Dropping Elephant	156
6	Sandworm	154
7	Naikon	127

['KERNEL32.dll',
'ADVAPI32.dll' ,
'HKCU\\SESSIONINFORMTION\\ProgramCount' ,
'WS2_32.dll' ,
'USER32.dll' ,
'C:\\WINDOWS\\explorer. exe' ,
'MSVCRT. dll']

Figure 2: A sample diagram of the behavior data of a malware sample.

operating system, such as "microsoft, documents, desktop." The words in this set will be automatically eliminated, and the TF-IDF value will no longer be calculated. Besides, filter out some data that appear too frequently and too much.

Finally, after the above calculations, the behavior data of the data set are represented as a feature matrix S, which includes more than one thousand features. In addition, we analyzed the top 20 features TF-IDF value size in each organization and the size distribution of some features is different in each organization, indicating that our feature representation is effective, and this method can obtain some features with a degree of difference (see Figures 4–10).

2.2. Feature Dimensionality Reduction. Since the feature representation generates many feature dimensions and sparse feature vector values in the previous step, dimensionality reduction of feature vectors is a more feasible method to increase speed and efficiency of detection and improve the model fitting effect. Here, the chi-square test is used for feature dimensionality reduction. The chi-square test (CHI) is also called x^2 statistic, which is used to test the independence of two variables. The chi-square test feature selection algorithm is mainly used to determine the correlation between the feature item t_i and the category c_j. Time obeys the x^2 distribution, and the value x^2 reflects the degree of correlation between t_i and c_j. If a word has a higher x^2 relative to a certain category, it indicates that the word has a great correlation with that category. The calculation method is shown in

$$x^2\left(t_i, c_j\right) = \frac{N \times (A\ D - BC)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)}. \quad (4)$$

Among them, A means the number of texts belonging to the category c_j and containing the feature item t_i; B means the number of texts containing t_i but not belonging to c_j; C means the number of texts that do not contain t_i but belong

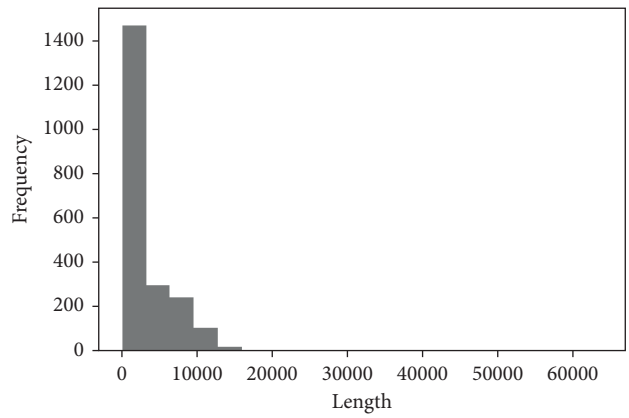

Figure 3: The length histogram of the sample's behavior text data.

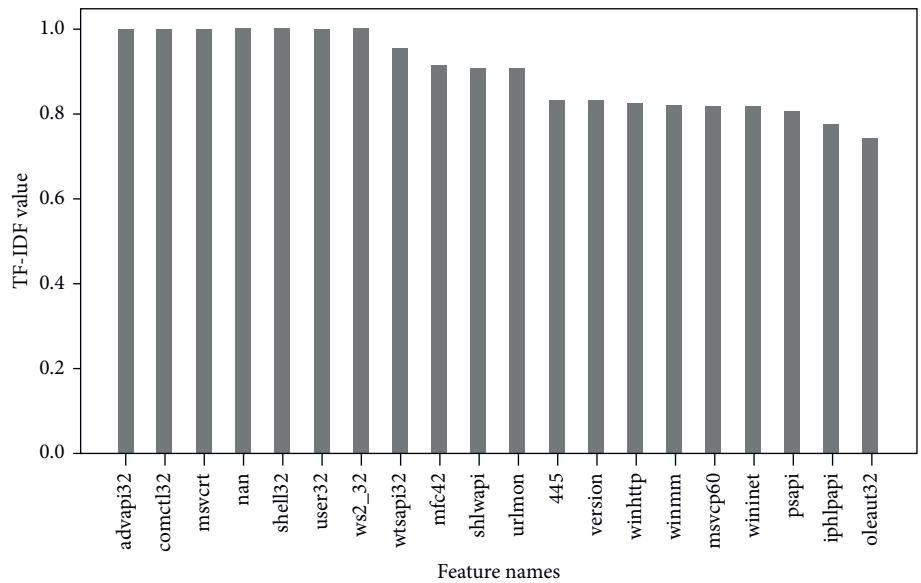

Figure 4: The top 20 features of the Lazarus group sample.

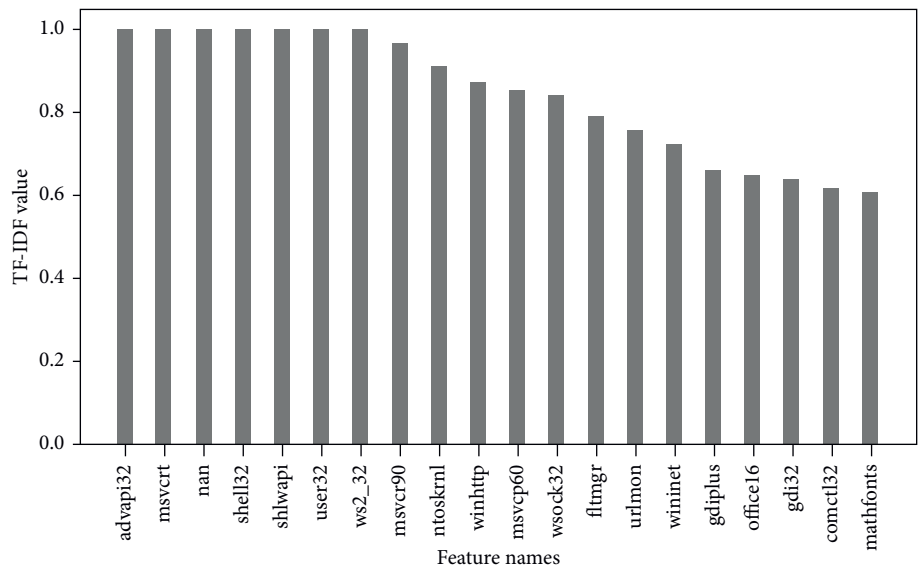

Figure 5: The top 20 features of the APT28 sample.

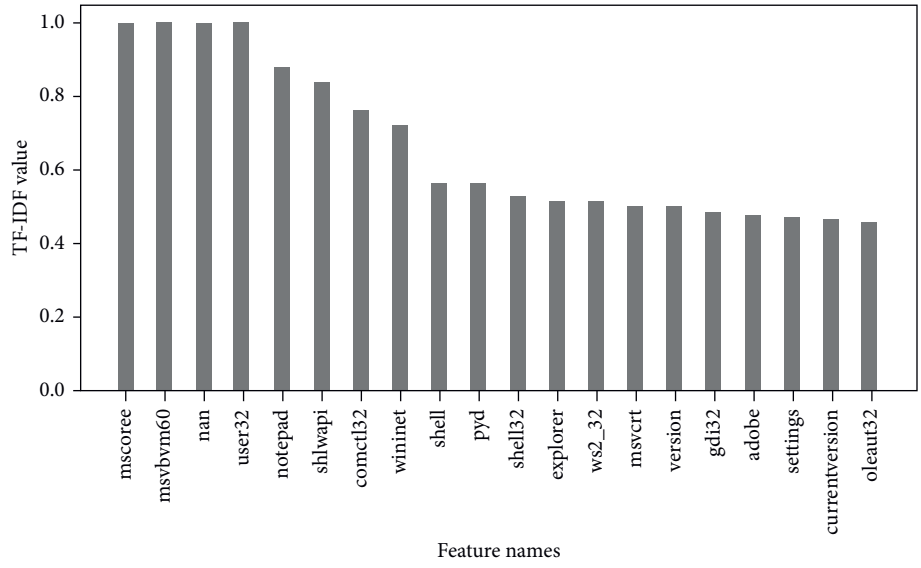

Figure 6: The top 20 features of the Operation C-Major sample.

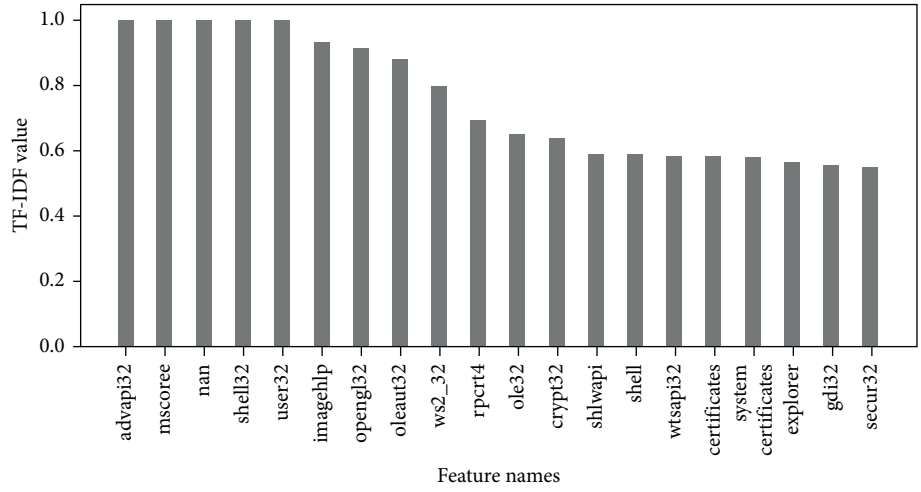

Figure 7: The top 20 features of the APT29 sample.

to c_j; D represents the number of texts that do not belong to c_j and do not contain t_i feature items, and $N = A + B + C + D$. The feature items selected by the chi-square test have a strong correlation with the text category, and the feature items represent category information. According to the calculation formula, the first k words of each category are selected as features according to certain requirements. Calculate the x^2 statistics between each feature and the category standard, and finally select the k features with the highest score x^2.

We calculate the chi-square value of more than one thousand features generated after feature representation. The larger the chi-square value, the better the ability of the feature to distinguish the sample. The top 20 characteristics of chi-square value size are shown in Figure 11.

2.3. Classification Model. To deal with the trouble of unbalanced classification and multiclassification in APT data sets, this paper designs the SMOTE-RF model. The model integrates SMOTE and random forest algorithms. The SMOTE algorithm is a simple and effective oversampling method proposed by Chawla et al. [30]. This method randomly selects k nearest neighbors among minority samples and increases the number of minority samples through interpolation with k nearest neighbors to improve imbalanced data set distribution. Random forest is an integrated algorithm based on decision tree learners. It uses bootstrap resampling technology of the self-service method to randomly select k samples from the original training sample set N with replacement to generate a new training sample set. Some samples may be selected multiple times, and some may

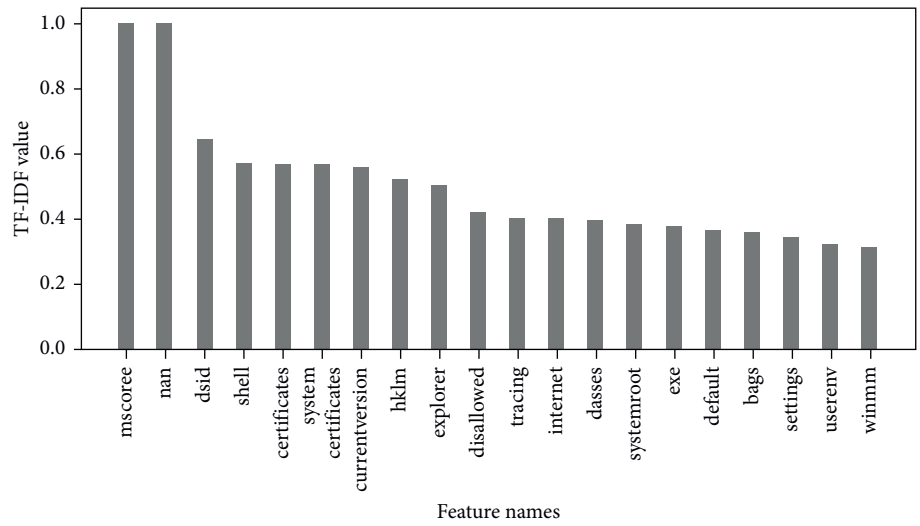

Figure 8: The top 20 features of the Dropping Elephant sample.

not be selected once and then generate k classification trees to form a random forest based on the self-service sample set. The final classification result is voted by all classification trees in the forest, and the algorithm has good generalization ability.

The SMOTE-RF model is first based on the number of samples N of the category with the largest number of samples in the data set S' and uses the SMOTE algorithm to generate N new samples for the samples of each other category. Then, multiclassification training is performed based on the random forest algorithm to obtain the classification model, and finally, the output category is predicted.

The SMOTE-RF model construction process is divided into seven steps.

The original training set is S_{train}', the class with the largest number of samples is N, other classes are classified as minority classes, the minority class sample set is M, i is a sample of the minority class, and its feature vector is $x_i, i \in (1, \ldots, M)$:

Step 1: calculate the Euclidean distance between each sample x_i in the minority M samples and the minority sample M. Get the k nearest neighbors of the sample.

Step 2: randomly select N samples from the k nearest neighbors, and each sample and its selected N nearest neighbor samples are combined into N new samples according to the following equation:

$$x_i(\text{new}) = x_i + \text{rand}\left(x_{ij} - x_i\right). \tag{5}$$

Among them, $x_i(\text{new})$ represents the newly added minority sample and x_{ij} indicates the j the nearest neighbor sample of x_i, where $j = 1, \ldots, N$.

Step 3: put the newly synthesized samples into the original training set S'_{train} to form a new balanced training set S'_{train}.

Step 4: use bootstrap resampling technology to randomly select T samples from all T samples in the new training set S'_{train}, and select T samples to train a decision tree.

Step 5: assuming that each sample has F features, when each node of the decision tree is split, randomly select f features ($f < F$) from these F features as candidate features and then select from candidate features. The feature that yields the best value splits the nodes of the decision tree.

Step 6: follow steps 4-5 to generate T decision trees to construct a random forest.

Step 7: vote the classification target through all the trees, and the classification with the most votes is the final classification result.

3. Experimental Results and Analysis

3.1. Model Evaluation Index. The experiment in this paper is a multiclassification. In order to comprehensively investigate various classifications, the performance indicators choose precision, recall, and $F1$-score. Also, a confusion matrix is used to represent the results of this classification, as shown in Table 2.

For the i-th category ($1 \le i \le n$), precision (P_i), recall rate (R_i), and F-score (F_score_i), respectively, are

$$P_i = \frac{c_{ii}}{\sum_j c_{ji}}, \qquad R_i = \frac{c_{ii}}{\sum_j c_{ij}}, \tag{6}$$

$$F\_\text{score}_i = 2 \times \frac{P_i * R_i}{P_i + R_i}. \tag{7}$$

Finally, the arithmetic average of the indicators of each category is calculated to obtain the macro average, which is used to measure the overall effect of each algorithm classification:

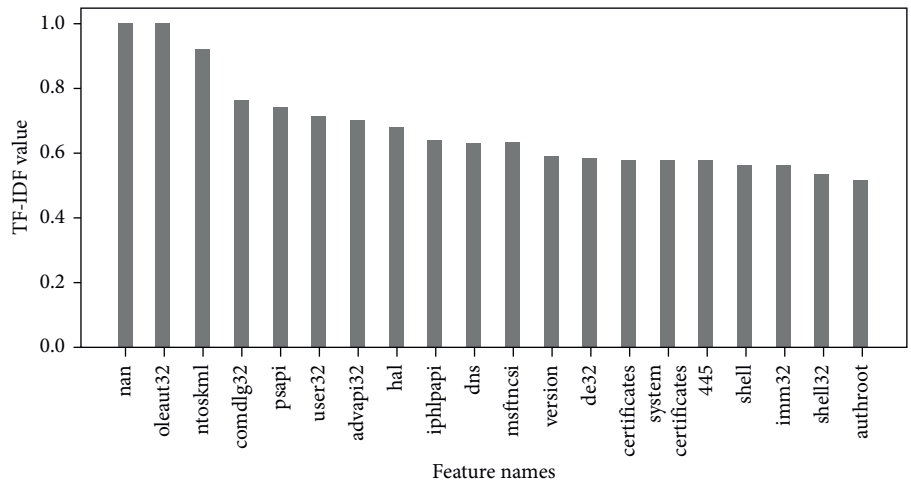

Figure 9: The top 20 features of the Sandworm sample.

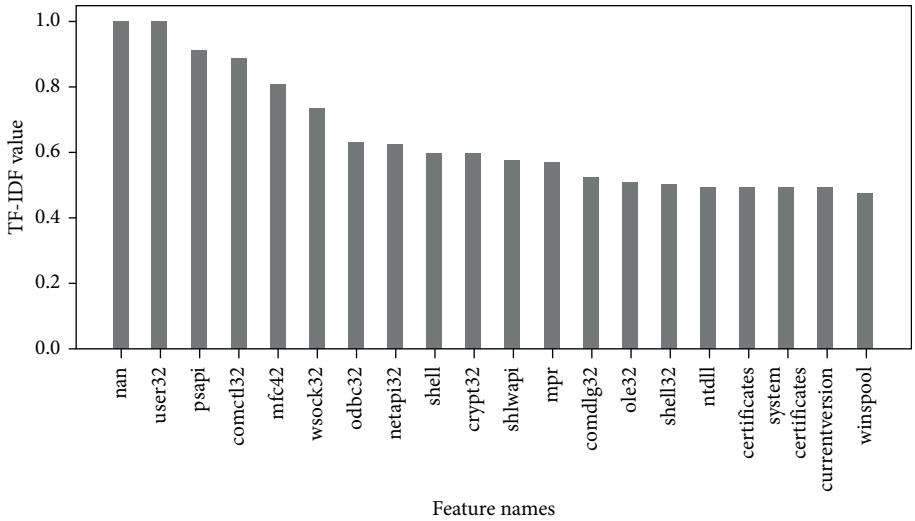

Figure 10: The top 20 features of the Naikon sample.

$$P_{\text{macro}} = \frac{1}{n}\sum_{i=1}^{n} P_i,$$

$$R_{\text{macro}} = \frac{1}{n}\sum_{i=1}^{n} R_i, \tag{8}$$

$$F\_\text{score}_{\text{macro}} = 2 \times \frac{P_{\text{macro}} * R_{\text{macro}}}{P_{\text{macro}} + R_{\text{macro}}}.$$

3.2. Experimental Results. To compare the prediction results, the algorithms that often perform well on classification tasks, such as KNN algorithm, DT algorithm, and XGBoost algorithm, are selected here and the SMOTE-RF model of this article is compared and verified by experiments. The prediction results of each model in each category are shown in Table 3. It can be seen that APT29, Dropping Elephant, and Sandworm have better classification effects on KNN, DT, XGB, and SMOTE-RF models, respectively. Operation Sandworm has the best classification effect on the SMOTE-RF model, with an F-score reaching 0.939. The classification effect of the Dropping Elephant organization on the DT model is the best, with an F-score reaching 0.903. The classification effect of Operation C-Major organization on the four models is the same. Lazarus, APT28, APT29, and Naikon organizations all achieved the highest F-score on the SMOTE-RF model. The F-score of Lazarus Group and APT 28 is relatively low. After data analysis, the main reason is that some samples have fewer text data, which leads to very few effective features extracted. In addition, combining the performance of each model on the training set (see Figure 12) and test set (see Figure 13), it can be seen that the DT model is the most unstable and the SMOTE-RF model has the best overall classification effect and stability. Our

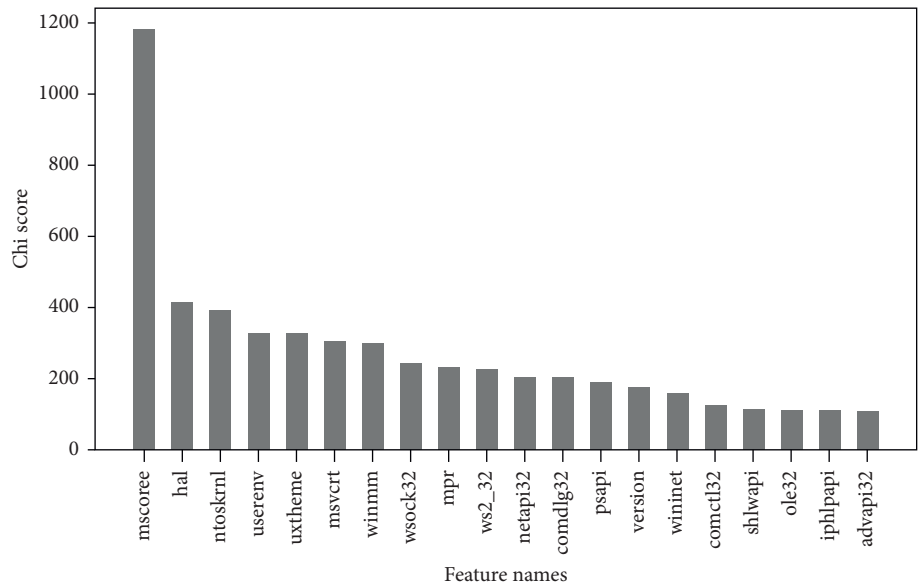

Figure 11: Top 20 features of the chi-square value.

Table 2: Confusion matrix representation of classification results.

True value	Predictive value		
	Group 1	Group 2	Group 3
Group 1	c_{11}	c_{12}	c_{13}
Group 2	c_{21}	c_{22}	c_{23}
Group 3	c_{31}	c_{32}	c_{33}

Table 3: Classification results of each model on each APT organization.

APT organization	Evaluation	KNN	DT	XGB	SMOTE-RF
Lazarus group	Precision	0.791	0.750	0.800	0.845
	Recall	0.507	0.493	0.478	0.567
	F-score	0.618	0.595	0.598	0.644
APT28	Precision	0.360	0.351	0.355	0.366
	Recall	0.854	0.833	0.792	0.854
	F-score	0.506	0.494	0.490	0.513
Operation C-Major	Precision	0.889	0.889	0.889	0.889
	Recall	0.828	0.828	0.828	0.828
	F-score	0.857	0.857	0.857	0.857
APT29	Precision	0.937	0.938	0.912	0.968
	Recall	0.825	0.839	0.857	0.834
	F-score	0.877	0.886	0.884	0.896
Dropping Elephant	Precision	0.927	0.980	0.944	0.927
	Recall	0.836	0.836	0.836	0.836
	F-score	0.879	0.903	0.887	0.879
Sandworm	Precision	0.840	0.917	1.0	1.0
	Recall	0.808	0.846	0.846	0.885
	F-score	0.824	0.880	0.917	0.939
Naikon	Precision	0.913	0.957	0.917	0.957
	Recall	0.700	0.733	0.733	0.733
	F-score	0.792	0.830	0.815	0.830

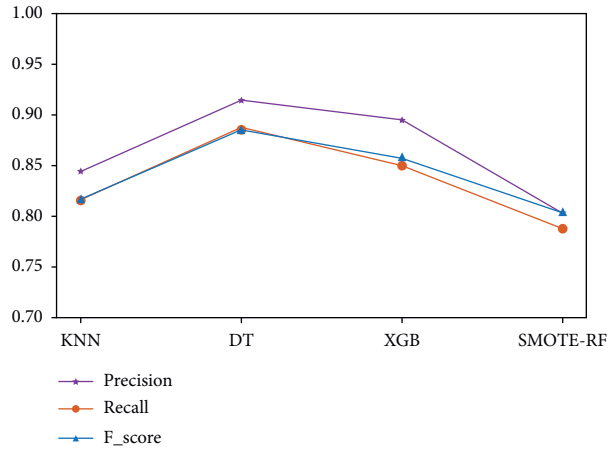

Figure 12: Performance indicators of each model on the train set.

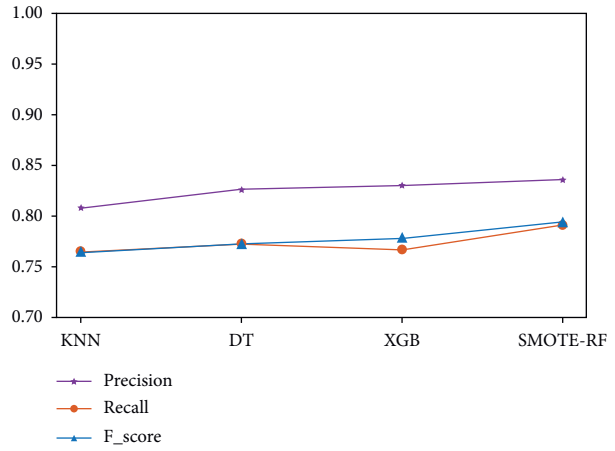

Figure 13: Performance indicators of each model on the test set.

experimental results prove the effectiveness of our feature extraction method and the superiority of our model.

4. Conclusions

In recent years, cyber-attacks are being used by various countries and intelligence agencies as one of the important means to achieve their political, diplomatic, military, and other purposes. The detection of APT has aroused widespread concern in information security and academic research circles. The classification of the attribution of APT malware samples is conducive to constructing attack scenarios, tracking attackers, and effectively identifying APT attack organizations of subsequent incidents. This paper proposes a classification method of APT organizations based on machine learning and malware. This method is based on the behavior data with APT organization tags obtained from dynamic analysis of APT malicious software acquired from the Internet of Things devices, and relatively strong feature vectors are obtained through feature representation and feature dimensionality reduction. Considering the sample imbalance in the data set, this paper designs a SMOTE-RF model that integrates SMOTE and random forest algorithms. Finally, the effectiveness of the proposed method for the attribution analysis of APT malware is verified by multiple sets of experiments. Among them, our method of feature extraction can achieve more than 80% accuracy in general models and the SMOTE-RF model performs well and has stable performance in the classification of APT malware. Next, we will combine non-APT malware samples to further study the features of APT attacks and each organization and to better identify APT attack activities and protect the security of the next generation of complex networks.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Shudong Li and Qianqing Zhang contributed equally to this work.

Acknowledgments

This research was funded by the Key R D Program of Guangdong Province (No. 2019B010136003), NSFC (Nos. 62072131 and 61972106), Science and Technology Projects in Guangzhou (No. 202102010442), National Key Research and Development Program of China (No. 2019QY1406), Open Project of National Engineering Laboratory for Mobile Internet System and Application Security, and Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2019). The authors thank the data provided by the NSFOCUS company.

References

[1] L. Xiao, X. Wan, C. Dai, X. Du, X. Chen, and M. Guizani, "Security in mobile edge caching with reinforcement learning," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 116–122, June 2018.

[2] H. Yang, S. Li, X. Wu, H. Lu, and W. Han, "A novel solution for malicious code detection and family clustering based on machine learning," *IEEE Access*, vol. 7, no. 1, pp. 148853–148860.

[3] I. Ghafir and V. Prenosil, "Advanced persistent threat attack detection: an overview," *International Journal Of Advances In Computer Networks And Its Security*, vol. 4, no. 4, pp. 154–158, 2014.

[4] T. M. Chen, "Stuxnet, the real start of cyber warfare? [Editor's Note]," *IEEE Network*, vol. 24, no. 6, pp. 2-3, 2010.

[5] W. Chen, X. Helu, C. Jin et al., "Advanced persistent threat organization identification based on software gene of malware," *Transactions on Emerging Telecommunications Technologies*, vol. 31, 2020.

[6] S. Li, D. Zhao, X. Wu, Z. Tian, A. Li, and Z. Wang, "Functional immunization of networks based on message passing," *Applied Mathematics and Computation*, vol. 366, Article ID 124728, 2020.

[7] A. S. Bist and S. Jalal, "Identification of metamorphic viruses [C]," in *Proceedings of the 2014 IEEE International Advance Computing Conference (IACC)*, pp. 1163–1168, Gurgaon, India, February 2014.

[8] "Going ATOMIC: clustering and associating attacker activity at scale," 2019, https://www.fireeye.com/blog/threat-research/2019/03/clustering-and-associating-attacker-activity-at-scale.html.

[9] F. Turkmen, S. Foley, B. O'Sullivan, W. Fitzgerald, T. Hadzic, and M. S. Basagiannis, "Explanations and relaxations for policy conflicts in physical access control," in *Proceedings of the 25th IEEE International Conference on Tools with Artificial Intelligence*, pp. 330–336, IEEE Press, Herndon, VA, USA, November 2013.

[10] R. Koike, N. Nakaya, and Y. Koi, "Development of system for the automatic generation of unknown virus extermination software," in *Proceedings of the 2007 International Symposium on Applications and the Internet*, Hiroshima, Japan, January 2007.

[11] S. Li, L. Jiang, X. Wu, W. Han, D. Zhao, and Z. Wang, "A weighted network community detection algorithm based on deep learning," *Applied Mathematics and Computation*, vol. 401, Article ID 126012, 2021.

[12] S. Shen, H. Zhou, F. Sheng et al., "HSIRD: a model for characterizing dynamics of malware diffusion in heterogeneous WSNs," *Journal of Network and Computer Applications*, vol. 146, 2019.

[13] S. Li, Y. Li, W. Han, X. Du, M. Guizani, and Z. Tian, "Malicious mining code detection based on ensemble learning in cloud computing environment," *Simulation Modelling Practice and Theory*, Article ID 102391, 2021.

[14] M. Fan, S. Li, X. Wu, W. Han, Z. Gu, and Z. Tian, "A novel malware detection framework based on weighted heterograph," in *Proceedings of the CIAT 2020: 2020 International Conference on Cyberspace Innovation of Advanced Technologies*, vol. 4-6, pp. 39–43, Guangzhou China, December 2020.

[15] P. Zhu, X. Wang, D. Jia, Y. Guo, S. Li, and C. Chu, "Investigating the co-evolution of node reputation and edge-strategy in prisoner's dilemma game," *Applied Mathematics and Computation*, vol. 386, Article ID 125474, 2020.

[16] G. Zhao, K. Xu, L. Xu, and B. Wu, "Detecting APT malware infections based on malicious DNS and traffic analysis," *IEEE Access*, vol. 3, pp. 1132–1142, 2015.

[17] M. Abomhara and G. M. Kien, "Cyber security and the Internet of Things: vulnerabilities, threats, intruders and attacks," *Journal of Cyber Security and Mobility*, vol. 4, no. 1, pp. 65–88, 2015.

[18] Y. Sung, P. K. Sharma, E. M. Lopez, and J. Park, "Fs-opensecurity: a taxonomic modeling of security threats in sdn for future sustainable computing," *Sustainability*, vol. 8, no. 9, 2016.

[19] M. Lee and D. Lewis, "Clustering disparate attacks: mapping the activities of the advanced persistent threat," 2013, https://www.virusbulletin.com/uploads/pdf/conference/_slides/2011/Lee-VB2011.pdfAccessed.

[20] S. Shen, L. Huang, H. Zhou, S. Yu, E. Fan, and Q. Cao, "Multistage signaling game-based optimal detection strategies for suppressing malware diffusion in fog-cloud-based IoT networks," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1043–1054, 2018.

[21] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "CorrAUC: a malicious bot-IoT traffic detection method in IoT network using machine learning techniques," *IEEE Internet of Things Journal*, vol. 8, 2020.

[22] S. Shen, H. Ma, E. Fan et al., "A non-cooperative non-zero-sum game-based dependability assessment of heterogeneous WSNs with malware diffusion," *Journal of Network and Computer Applications*, vol. 91, pp. 26–35, 2017.

[23] S. Yu, G. Gu, A. Barnawi, S. Guo, and I. Stojmenovic, "Malware propagation in large-scale networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 1, pp. 170–179, 2015.

[24] S. Peng, S. Yu, and A. Yang, "Smartphone malware and its propagation modeling: a survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 925–941, 2014.

[25] Y. Qiao, X. Yun, and Y. Zhang, "How to automatically identify the homology of different malware," in *Proceedings of the 2016 IEEE Trustcom/BigDataSE/ISPA*, pp. 929–936, Tianjin, China, August 2016.

[26] N. Moran and J. Bennett, "Supply chain analysis: from quarter master to sunshop," Technical Reporthttps://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-maware-supply-chain.pdf, Fire EyeLabs, 2013.

[27] http://www.freebuf.com/news/92945.html.

[28] E. Hutchins, M. Cloppert, and R. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare and Security Research*, vol. 1, 2011.

[29] "The MITRE Corporation, ATT&cK matrix[EB/0L]," 2020, https://attack.mitre.org/.
[30] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321–357, 2002.