International Journal of Digital Crime and Forensics
Volume 11 • Issue 4 • October-December 2019

A Highly Efficient Remote Access Trojan Detection Method

Wei Jiang, Beijing University of Technology, Chinese Academy of Cyberspace Studies, Beijing, China
Xianda Wu, Beijing University of Technology, Beijing, China
Xiang Cui, Guangzhou University, Guangzhou, China
Chaoge Liu, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

ABSTRACT

Nowadays, machine learning is popular in remote access Trojan (RAT) detection which can create patterns for decision-making. However, most research focus on improving the detection rate and reducing the false negative rate, therefore they ignore the result of abnormal samples. In addition, most classifiers select several proprietary applications and RATs as their training set, which makes them difficult to adapt to the real environment. In this article, the authors address the issue of imbalance dataset between normal and RAT samples, and propose a highly efficient method of detecting RATs in real traffic. In the authors method, they generate eight features by combining the size, the inter-arrival and the flag from one packet sequence. Then, they preprocess the imbalance dataset and implement a classifier by XGBoost algorithm. The classifier achieves a false negative rate of less than 0.18%. Moreover, the authors demonstrate that their classifier is capable of detecting unknown RAT.

KEYWORDS

Feature Extraction, Machine Learning, Network Behavior, Remote Access Trojan, Traffic

1. INTRODUCTION

Remote Access Trojan (RAT) is a malicious tool for attackers to do remote control and intercept information, which causes serious impacts and huge losses to the states, enterprises and individuals. Typically, the RAT consists of control side and controlled side. Attackers can use the method of spear phishing and social engineering attack to find the machines which can be infected, and then adopt the standard TCP/IP or UDP protocol to achieve real-time communication between the control and the controlled side. Not being same with the traditional security threats, the RAT is often used in data theft and privacy snooping with the performance of full feature, concealment and long persistence. Generally speaking, one of the hide methods is inject themselves into the other legal process, so it does not display on the task list. In addition, the operations of RATs are gradually similar with legal applications, which make the detection of RAT more difficult.

The different RATs have widely difference on setting function and operating environment, but their network behavior has a certain similarity. Therefore, the network traffic feature can be extracted to train the detection model. We can take this traffic as an indicator, which reflected the integration of all information between the control side and the controlled side. The abnormal network condition can be reflected by the parameter of traffic. Based on this, we present a highly efficient RAT detection classifier which allows us to detect the potential communication of RATs in the hybrid traffic. Much

DOI: 10.4018/IJDCF.2019100101

like the other researches, we collect the traffic from twelve hosts, which included 22 kinds of publicly available RATs and at least 10 kinds of known normal application software. The normal application software includes TeamViewer, Thunder, BitComet, Bit Torrent, Chrome, PPTV, QQ, Sun, WeChat, and so on. Then, we use some RAT traffic to verify the ability of our classifier on discovering the unknown RAT, these samples are collected from two parts: internet and our lab.

In summary, this article has the following contributions:

We extract eight features from the RAT's flows and sessions. In this phrase, we set a threshold for each flow, which ensures that our detection in a controlled time with more efficiency.

We select the hybrid traffic as our dataset and solve the problem of imbalanced dataset. From the contrastive experiment, we demonstrate that after we process the problem of imbalanced dataset, the false positive rate has been reduced.

As far as we know, this is the first time that the XGBoost classification algorithm (Chen & Guestrin, 2016) is used to detect the RAT. Additionally, we witness that our method has the ability to detect some potentially unknown RATs.

We evaluate our classifier by real-world traffic. Our method achieves good performance with high accuracy and low false negative.

The remainder of the paper is organized as follows: Section 2 discusses the detection methods of RAT which are related to the recent studies. In section 3, we introduce our method, describe the course of RAT network features selection and illustrate the realization of our method. Section 4 describes the experiment and evaluate it results. Finally, we summarize the whole article and put forward the future work in section 5.

2. RELATED WORKS

Based on the difference of RAT detection technologies, the detection circumstance can be divided into three parts: based on the host, network-based detection (Adachi & Omote, 2016, Chawla et al., 2002, Chen & Guestrin, 2016, Fukushima et al., 2010, Liu et al., 2006) and the hybrid measure.

2.1. Host-Based Detection

Host-based detection is one of the earliest technologies, which includes the signature detection (Deng et al., 2003), heuristic scanning (Sanok, D. J. 2005) and host internal behavior analysis (Xiang et al., 2009). The detection based on signature has a high accuracy, but it is difficult to detect the kind of unknown and variant. The latter two methods need to be in a controlled environment, monitoring the sensitive operation for determine whether it is a malicious sample. Niu, Liu, & Duan, (2014) extracted the call sequence of API function from the PE file, they introduced the attack tree. In their method, the call sequence of API function was matched against the attack tree, and they used the attack root node to represent the risk index of the event, and estimated the level of similarity to the Trojan; Liu et al. (2009) proposed a prototype. Their prototype searched the important file which contains the user's confidential information on the disk. And then, these files were monitored to found which processes accessed them by capturing and reflecting the IRPs. Detecting the behavior of host can achieve a better result, however, this method takes up more resources, and if some hybrid attacks split one function into multiple actions, it may be evaded.

2.2. Network-Based Detection

Network-based detection technology is a more popular method. It mainly analyses the suspicious network behavior, network protocol and network traffic. Compared with the host-based detection, the network-based traffic detection technology can aware the risk of the entire LAN, with lower deployment cost and better protection. In this area, content-based detection extract information from packets, which takes more time. With the improvement of hybrid encryption technology, the detection accuracy is reduced. Behavior-based detection is a means of improving network security. It does

not depend on the content of the packet, but on analyzing the influence of network environment on network behavior. This method is conducive to find out the unknown threats. Recently, detection by machine learning algorithm is more popular (Jiang & Omote, 2015, Wei et al., 2015, Li et al., 2012, Pu et al., 2013, Pallaprolu et al., 2017, Wu et al., 2016, Xu et al., 2015, Yamada et al., 2015, Zhang, Tong, & Qin, 2016), which includes KNN, C4.5, K-Means, Decision Tree, Random Forest and so on. Researches generated various detection classifiers by extracting the different network features. The performance of these classifiers mainly lies in the selection of datasets and eigenvalues. Recently research datasets can be divided into two kinds: RAT's traffic with the traffic of normal applications or RAT's traffic with a large quantity of normal network traffic. These two scenarios may lead to unbalanced data problems that affect the final detection effect.

2.3. Hybrid-Based Detection

In order to achieve a better detection result, some researchers combined host-based features with network-based features to find the threat in the network. The first systematic study of the RAT's behavior was generated by Farinholt et al. (2017). Their research was divided into two parts, the distribution statistics of DarkComent attackers and the behavior of attackers. In the behavioral studies, they monitored the behavior of the RAT on the host, and deployed a comprehensive set of network signatures to match the decrypted traffic. The follow-up study refers to the analysis of RAT behaviors. The network behavior identification in our paper refers to detecting RAT without decrypting the flow content. Process number is one of the most important features in hybrid measure (Liang et al., 2013, Peng et al., 2012, Wu et al., 2006). Adachi & Omote, (2016) extracted some features from processes and sessions, which reduced the FNR value to zero and ensured a high accuracy. Although the hybrid method has a very good detection results, the number of samples still has room to be upgraded, and the detection of hybrid flow needs more investigations.

Based on the above analysis, we designed a detection method to solve the problem of data imbalance, and completed the hybrid flow detection.

3. DETECTION METHOD

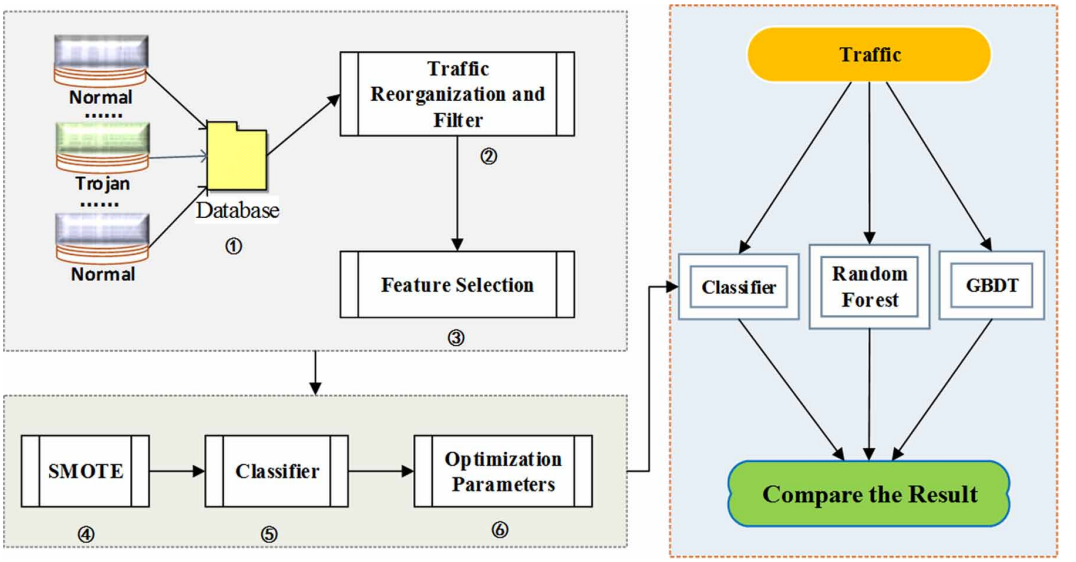
The activities of RAT can be summarized as follows: the implant phase, the install phase, the start-up phase, and the network communication phase. We focus on the network communication phase. As the RAT has a too long communication cycle, if we detect it at the beginning of the attack and finished at the attack stop, it will lead to long time consumption. Therefore, the goal of our method is that as long as the attacker establishes a connection with the target, we can find the attack in a controlled time. We consider it's a classification problem for both types of samples (normal and RATs). Our method consists of three main phrases: traffic collection and feature extraction, establishing and optimizing the classifier, and abnormal detection. The overall detection framework is shown in Figure 1.

3.1. Traffic Collection and Feature Extraction

This article is devoted to detecting RAT, the most important issue is to choose some effective network features to recognize the RAT (Table 1) and normal traffic. To better understanding our work, it is important for us to give the definition of the Flow, Session and Periodic Flow:

***Definition 1**: [Flow].* For the traffic sample, we select the traffic based on the TCP protocol, and extract the "flow" based on the different source IP addresses and destination IP addresses. The flows in this paper begin with the three-way handshake starting with flag "SYN" until the communication time reaching threshold T. The total length of the flows is denoted as N.

Figure 1. Overall the detection framework



Definition 2: *[Session].* Session is formed by flow reorganize and filter. Each flow can be decomposed into 1 to n different [source IP address, source port, destination IP address, destination port] communication "Session".

Definition 3: *[Periodic Flow].* The "interval time" between two adjacent packets is defined as t, $T_{Linternal}$ is represented the interval of all packets are collected in one flow, and is written as $T_{Linternal}$= $\{t_0,t_1,t_2.....t_{N-1}\}$; The sum of elements in $T_{Linternal}$ is denoted as the total time, which expressed by SUMT; the flow of the whole time T is called "periodic flow".

This process can be split into two steps: traffic collection and feature extraction.

3.2. Traffic Collection

As is shown in Figure 1, step ① and ② represent the phrase of traffic collecting and filtering respectively. To prevent the sample from being highly similar, seven kinds of RAT traffic from the website Nuclear-EK-traffic (2014), Mila Parkour (2013) as our test sample. The other RAT traffic samples are collected from a controlled test environment, eighteen of them as train samples, four of them as test samples. In data collection stage, we use NetAnalyzer and Wireshark software to capture the communication flow from seven computers which in a controlled environment (two of which are implanted in Trojans 2 * and 3 *). These traffics can be divided into three categories: The first one includes 18 kinds of RATs flow, the second contains 10 kinds of known begin application software flow, and the third one comprises hybrid network flow. Eventually we collect the communication flow with a total of 291.17 hours (Table 2), which are stored as the file format of .pcap.

We do not consider the flows whose communication time is less than 1S. After filtering the traffic samples, we obtain 1862 flows as the training set T_1, of which 119 flows from RATs traffic. We select 70% of the data in T_1 as train dataset randomly, record as T_{R1}, the remained 30% used as test dataset, recorded as T_{E1}; In addition, the test set T_{E2} is the flow from the other five machines which total of 145.83 hours (Table 3). After filtering, we obtain 1342 flows, of which 86 flows belong to RAT. The test set is generated by the same method as the train dataset, in which 4 kinds of RAT (Bozok, HAKOPS RAT, Xtreme RAT and Comet Rat) not in train set.

3.3. Feature Extraction

In reality, traffic is more complex. If we group the traffic by flow, we cannot identify the short-term RAT flow timely from the long interaction between two IPs. If we group the traffic by session, the amount of data is relatively large, which effects the efficiency of our detection. In the view of the above issues, we analyze the connection features of the RAT's life cycle, set a time threshold for the traffic between the two IPs, and sum up the features of the different sessions, to prepare for the follow-up feature selection. Considering the time threshold set by Li et al. (2012), we set T to 5 minutes which in definition 1. Step ③ represents the process of feature processing. We extract network features of normal and RATs. Although many features can be calculated, not all of the features can achieve good detection results. Through the previous analysis, we combine the features from flag, time, packet number, number of bytes and the number of sessions in each flow. Finally, we decide to extract the following eight features to train our classifier. These eight network behavior features include six flow features and two session features:

3.3.1. Flow Features

The asymmetry in traffic is the most obvious performance of the RAT and normal applications. From the achievement of Farinholt et al. (2017), we can see that the remote desktop control is a common tool

Table 1. RAT sample and their versions

RAT Sample	Number of versions	RAT Sample	Number of versions
Nuclear	3	Gh0st	2
Bandook	1	Sx	1
Huigezi	1	DarkComent	2
Bozok	1	remote	1
CyberGate RAT	1	Taidoor	1
Pandora RAT	1	PoisionIvy	2
Comet Rat	1	SpyNet	1
Star RAT	1	Xtreme RAT	2
PcShare	1	njRAT	3
VanToM RAT	1	Plugx	2
X RAT	1	HAKOPS RAT	1

Table 2. Each machine sampling time for train dataset

Time(h)	25.34	18.91	145.32	2.0	0.85	96.5	2.25
Total time(h)	291.17						

Table 3. Each machine sampling time for test dataset

Computers	1	2	3	4	5
Time(h)	7.41	7.42	126.51	2.25	2.24
Total time(h)	145.83				

used by attackers. This is a commonly attack means and the traffic is difficult to cover up. Figure 2 and Figure 3 indicate the communication flow from a well-known RAT in a controllable environment. When the attacker makes a request, the target machine turns the results of the command, may be the system information, screen shots, audios, videos, etc., these data are much larger than the command data send from the attacker. From the performance of the flow, some behaviors can be reflected from the size of upstream and downstream traffic, the number of large data packets, and the transmission time, etc. Even if an attacker hides their big packet by split it into some small packets, these statistical features still can identify abnormal behavior effectively.

3.3.2. Session Features

Most of RATs will start a number of flows for information transmission in their communication phrase, such as: Nuclear, Darkcoment, PcShare and so on. These sessions can be divided into two categories: main connection and secondary connection. The main difference between the main connection and secondary connection is the connection time. Normally, main connection is established at the beginning of the communication, which is mainly responsible for sending commands and controlling information; Secondary connection is established based on the capabilities of RAT, such as: steal the keyboard records, access to remote screen information, etc., this kind of connection is ended after capabilities completion. The course of main connection and secondary connection are beneficial for RAT to hide itself. It is a common means to enhance the self-survival ability of RAT. In feature extraction phrase, we choose the number of sessions as one of our features. Assuming the longest session represent the main connection, we calculate the variance of the upstream packet to reflect the evenly spread of packets. The command control packets of RAT in the long connection are more than the normal application. We can see more about our features in Table 4.

3.4. Establish and Optimize the Classifier

At this stage, the XGBoost (eXtreme Gradient Boosting) algorithm and the SMOTE algorithm are applied to the RAT detection. The hybrid algorithm can deal with the problem of imbalance data and achieve a better result. The imbalance dataset can be interpreted as a large difference in the proportion

Figure 2. The number of bytes sent by the attacker

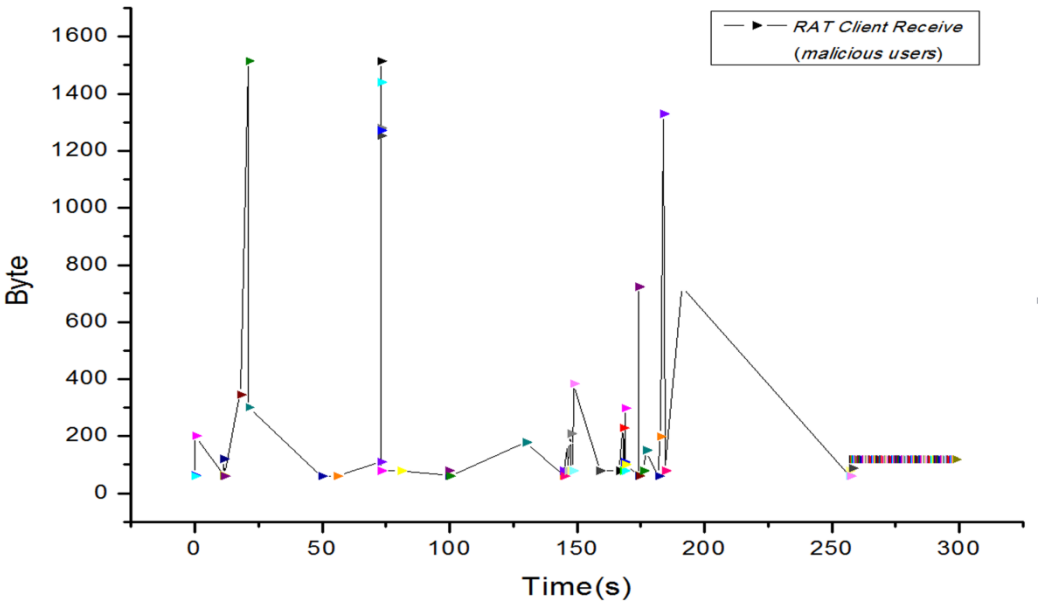
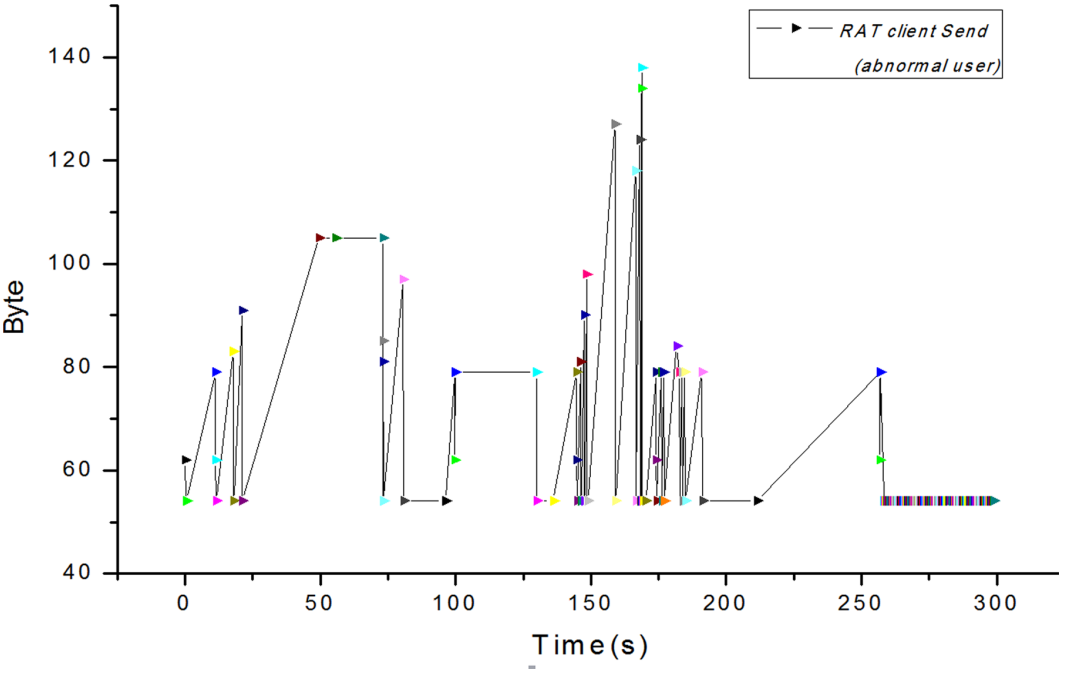


Figure 3. The number of bytes sent by the user



of certain classes of a sample relative to the other classes, and a small proportion of the dataset in the total sample. In this phrase, we perform some operations on the raw data sets and obtain new synthetic samples that increase the number of small samples. The main idea is to find K samples nearest to each of the RAT samples X in the original RAT data, and then randomly select N samples in these nearest neighbor datasets. Subsequently, a synthetic sample is inserted between the raw sample data and its neighbor samples. The procedure of this method is as follows:

For each RAT sample x, calculate the standard Euclidean distance with all of the minority class samples and find out its k neighbors. According to the imbalance proportion from the sample, we set a sampling rate indicated by N. We selects k random neighbors for each RAT samples x and assumes the selection of neighbors is x_n.

For each random sample x_n, a new sample is constructed according to the formula (1). x_{new} represents the new instance; x stands for a RAT example; y [i] is the first I near sample of X.

$$x_{new} = x + rand(0,1) * \left(y\left[i\right] - x\right) \tag{1}$$

Get new dataset. We process the T_{RI} data by SMOTE algorithm, the proportion of the begin flow and the RAT in T_{RI} from 1214: 89 to 1214:1246. The new synthetic sample is defined as $T_{synthesis}$.

Then ⑤⑥ represent that we train the XGBoost classifier after the process of the generated balanced dataset.

3.5. Abnormal Detection

This stage is to verify the effectiveness of the proposed method. In this section, we compare and analysis the results of random forest and GDBT algorithm with our method according detection the same dataset.

Table 4. Selected Features

Feature	From	Property	Symbol/Formula
MinPush	Flow	The average size of each packet which flag is[ACK, PUSH] in upstream minus the average size of each packet which flag is [ACK, PUSH] in downstream. In the time interval (0, T), the sum of bytes in the packet of the upstream which flag is [ACK, PUSH] is represented by P_{bup}, and the number is C_{bup}; the sum of bytes of the package whose downstream flag is [ACK, PUSH] expressed by P_{bdown}, and the number is C_{bdown}.	$\begin{cases} \frac{P_{bup}}{C_{bup}} > \frac{P_{bdown}}{C_{bdown}}, MinPush = 1 \\ \frac{P_{bup}}{C_{bup}} = \frac{P_{bdown}}{C_{bdown}}, MinPush = 0 \\ \frac{P_{bup}}{C_{bup}} < \frac{P_{bdown}}{C_{bdown}}, MinPush = -1 \end{cases}$
UpByte		The average number of bytes per second. The total numbers of bytes of all downstream packets in T time are expressed in P_{down} and the total time used for downstream packets in T time is denoted by T_{down}.	$UpByte = \frac{P_{down}}{T_{down}}$
InPac		The number of downstream packets sent per second. C_{down} indicates the number of all downstream packets, the total time used for downstream packets in T time is denoted by T_{down}.	$InPac = \frac{C_{down}}{T_{down}}$
Byte Value		The average number of bytes per second in upstream divide the average number of bytes per second in downstream. According to $T_{Linternal,}$ the total time taken for the upstream packet in T time is T_{up}, and the total number of bytes in the upstream packet is P_{up}.	$ByteValue = \frac{P_{up}.T_{down}}{T_{up}.P_{down}}$
SendMax		The number of packages whose size are greater than 90.	
SunPac		The number of packets, which flag contains [FIN,ACK] or [RST, ACK]	
Session	Session	The number of session.	
SesVar		The variance of all the upstream packets from the longest session.	

4. EXPERIMENTS AND RESULTS

In this section, we introduce the parameters to evaluation our classifier, and compare the results of different methods.

4.1. Evaluation Measures

For the imbalanced dataset detection, there are two criteria: Confusion Matrix and G-Mean. In the confusion matrix, each line represents the forecast and each column represents the actual category, TP, FP, FN, TN. The correctness and F values can be obtained by confusing the matrix. According to the confusion matrix we can get the Accuracy and F value. The formula (2) (3) (4) respectively stands for Accuracy, Recall and F-Measure.

$$Accuracy = \frac{TP + TN}{P_C + N_C} \tag{2}$$

$$\mathrm{Re}\,call = \frac{\mathrm{T}P}{\mathrm{T}P + \mathrm{F}N} \tag{3}$$

$$F\text{-}Measure = \frac{(1+\beta)^2.\mathrm{Re}\,call.\mathrm{Pr}\,ecision}{\beta^2.\mathrm{Re}\,call.\mathrm{Pr}\,ecision} \tag{4}$$

For the imbalanced data, even if all of the prediction results are the most classes, a lower error rate can be achieved with a higher accuracy rate, while the F value combines the accuracy and recall rate of a few classes, so it can measure the performance of the classifier under the imbalanced data, the parameter β is usually 1. G-Mean is another indicator to evaluate the imbalanced data. To various samples, it is the geometric mean of the accuracy. The larger the value is, the performance of the classifier is better. The formula is shown in (5).

$$\mathrm{G}\text{-}mean = \sqrt{\frac{TN}{TN+FP} \times \frac{TP}{TP+FN}} \tag{5}$$

4.2. Result and Analysis

In this section, we adopt the Grid Search method to determine the best parameters by cross validation, and then used these parameters to train a classifier. Then, we compare the random forest classifier, GBDT classifier and our classifier on $T_{synthesis}$. Additionally, in order to prove the validity of our classifier, we compare our classifier with the XGBoost classifier which is trained by original dataset and this classifier are tested by T_{R2} and $T_{synthesis}$. We implement our model by Python. In order to effectively avoid overfitting and the lack of learning state, we conduct a K-fold cross validation. In this paper, the cross-validation of the three classifiers is set to 6. Table 5 shows the results of SMOTE + Random Forest, SMOTE + GBDT and SMOTE + XGBoost with 6-fold cross-validation. Figure 4 shows the four indicators of the cross-validation of our training phase, and the last four bars represent the average parameters after the six cross-validation.

Subsequently, we compare the detection results of three classifiers on the test set T_{E1}. In T_{E1}, we can see the number of flow is 559, of which 30 belong to RAT. For the test set T_{E2}, samples are more abundant than T_{E1}. In T_{E2}, we can see the number of flow is 1342, of which 86 belong to RAT. T_{E2} contains four kinds of Trojans that are not available in the training set and part of RAT traffic from internet. Table 6 shows the results of the XGBoost algorithm and the other two algorithms, respectively, for datasets T_{E1} and T_{E2}. Compared with SMOTE + RandomForest, the accuracy of our classifier exceeded 0.54%, the recall has increased by 3.33%, F_1 has increased by 4.81% and GMean has increased by 3.26%. Similarly, compared with SMOTE + GBDT, the accuracy of our classifier exceeded 0.89%, and for the important detection parameters of imbalance data, the recall of our classifier increased by 10%, F_1 increased by 8.36%, GMean increased by 8.37%.

We can see that the method in this paper can find out all RATs in T_{E1}, and the evaluation criteria (*) are higher than the other two detection models. In T_{E2}, the number of begin samples and RAT samples are more than T_{E1}, the overall test results are shown in Table 7. In this table, compared with SMOTE + Random Forest, although the accuracy rate has a small difference of 0.75%, the recall of our classifier improved 6.98%, F_1 increased by 5.85%, GMean increased by 2.64%. It is obvious that the performance is better. Similarly, compared with SMOTE + GBDT, although the two methods have the same accuracy, for the important detection parameters of imbalance data, the recall of our

Table 5. Comparison of classifier's evaluation measures with 6-fold cross-validation

Detection Method	Accuracy (6CV)*	Recall (6CV)*	F1 (6CV)*	AUC (6CV)
XGBoost	98.46%	87.70%	88.65%	98.35%
SMOTE+RandomForest	99.03%	98.87%	98.95%	99.88%
SMOTE+GBDT	98.98%	99.20%	99.00%	99.92%
SMOTE+XGBoost	99.06%	98.79%	99.06%	99.92%

classifier increased by 2.33%, F_1 increased by 2.31%, GMean increased by 1.24%. It is proved that our method can detect the unknown RAT and make the model better than the SMOTE + Random Forest and SMOTE + GBDT algorithms when choosing a new dataset.

Table 8 shows the results of our classifier generated by the $T_{synthesis}$ dataset and the XGBoost detection model trained by the initial T_{R1} on the same dataset. From the table we can see that although the detection model generated by XGBoost algorithm has good detection effect, our method effectively reduces the false negatives rate of small sample's (RAT). In general, the classifier detection performance is efficient.

5. CONCLUSION

In this work, we analyzed eight features of RATs, and proposed a novel RAT detection method. The proposed method is focused on selecting some efficient traffic features to distinguish known and unknown RAT. Although their behavior has been changed, their network statistical features are difficult to change. That is exactly why our model is more stable. For the particularity of RAT detection in hybrid network traffic, we solved the problem of imbalance dataset. Subsequently, we implemented an efficient classification algorithm on the synthetic dataset. After two rounds of experiment, the result validate that our classifier can greatly reduce the false negative rate of small samples which trained on the equilibrium dataset, and prove the features of our selected and the first time of two algorithms for Trojan detection are very efficient. However, when the attackers use the other protocols such as UDP, our method will be effectiveness. Thus, we have to expand the proposed method on detecting the other kinds of protocols in the future. Furthermore, we will collect more samples of the RAT to strengthen our model. We expect the optimized model can not only deal with pcap file, but also implement the real-time detection.

ACKNOWLEDGMENT

The work was supported by the Ministry of Science and Technology of China (No.2016QY08D1602), National Development and Reform Commission Information Security S-pecial Fund (NDRC High-Tech No.[2015] 289). This study is also supported by Beijing Natural Science Foundation under Grant No.4172006, and General Program of Science and Technology Development Project of Beijing

Table 6. Detection results for test set T_{E1}

Detection Method	Accuracy*	Recall*	F1*	G-Mean*
SMOTE+RandomForest	99.28%	96.67%	93.55%	95.11%
SMOTE+GBDT	98.93%	90.00%	90.00%	90.00%
SMOTE+XGBoost	99.82%	100%	98.36%	98.37%

Figure 4. The results of our classifier training phase after per double cross validation

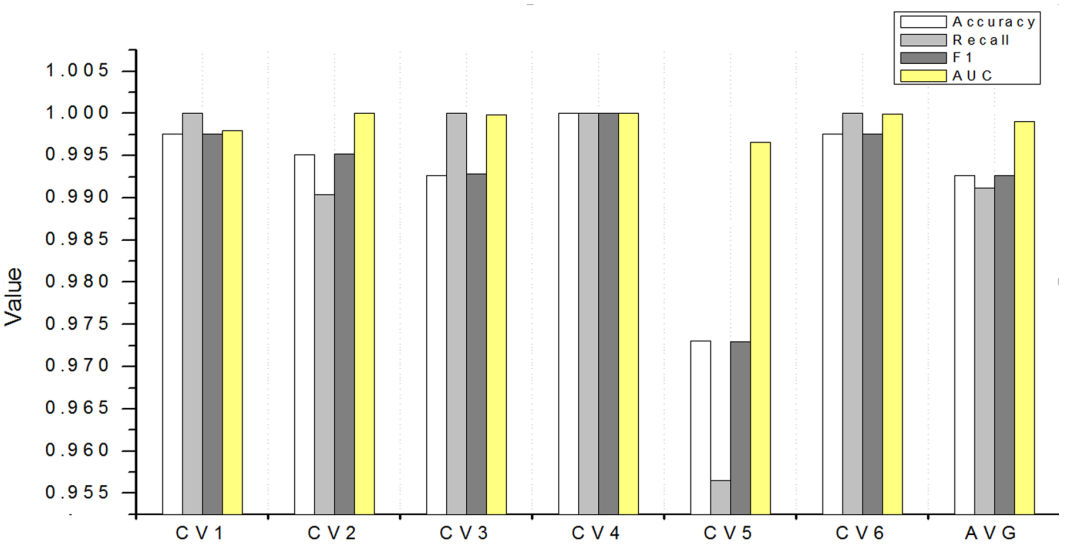


Table 7. Detection Results for Test Set T_{E2}

Detection Method	Accuracy*	Recall*	F1*	G-Mean*
SMOTE+RandomForest	99.18%	93.02%	93.57%	96.78%
SMOTE+GBDT	99.63%	97.67%	97.11%	98.18%
SMOTE+XGBoost	99.93%	100%	99.42%	99.42%

Table 8. Comparing the proposed classifier with XGBoost classifier

Dataset	T_{E1}		T_{E2}	
Detection Method	**XGBoost**	**SMOTE+XGBoost**	**XGBoost**	**SMOTE+XGBoost**
Accuracy*	99.82%	99.82%	99.85%	99.93%
Recall*	96.67%	100%	97.67%	100%
F*	98.31%	98.36%	98.82%	99.42%

Municipal Education Commission of China under Grant No.km201410005012, the Key Laboratory of Network Assessment Technology, Chinese Academy of Sciences and the Beijing Key Laboratory of Network security and Protection Technology.

REFERENCES

Adachi, D., & Omote, K. (2016). A host-based detection method of remote access trojan in the early stage.

Binru, N., Peiyu, L., & Linshan, D. (2014). An improved attack tree-based Trojan analysis and detection. *Computer Application and Software*, *29*(3), 277–280.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*(1), 321–357. doi:10.1613/jair.953

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM Sigkdd international conference on knowledge discovery and data mining* (pp. 785-794). ACM.

Deng, P. S., Wang, J. H., Shieh, W. G., & Yen, C. P. (2003). Intelligent automatic malicious code signatures extraction. In *IEEE, 2003 International Carnahan Conference on Security Technology Proceedings* (pp. 600-603). IEEE.

Farinholt, B., Rezaeirad, M., Pearce, P., Dharmdasani, H., Yin, H., & Blond, S. L. (2017). *To Catch a Ratter: Monitoring the Behavior of Amateur DarkComet RAT Operators in the Wild. In Security and Privacy* (pp. 770–787). IEEE.

Fukushima, Y., Sakai, A., Hori, Y., & Sakurai, K. (2010). *A behavior based malware detection scheme for avoiding false positive. In Secure Network Protocols* (pp. 79–84). IEEE.

Jiang, D., & Omote, K. (2015). An Approach to Detect Remote Access Trojan in the Early Stage of Communication. In *International Conference on Advanced Information NETWORKING and Applications* (pp. 706-713). IEEE. doi:10.1109/AINA.2015.257

Jinlong, W. U., Xiaofei, S. H. I., Jia, X. U., & Jun, S. H. I. (2016). *Hierarchical Detection of Trojan Behavior based on Random Forest*.

Li, S., Yun, X., Zhang, Y., Pang, Y., & Yin, T. (2012). A novel approach of detecting Trojan based on network behavior analysis.

Liang, Y., Peng, G., Zhang, H., & Wang, Y. (2013). An Unknown Trojan Detection Method Based on Software Network Behavior. 中国可信计算与信息安全学术会议, 18, 369-376).

Liu, R., Liu, E., Yang, J., Li, M., & Wang, F. (2006). *Optimizing the Hyper-parameters for SVM by Combining Evolution Strategies with a Grid Search. In Intelligent Control and Automation*. Springer.

Liu, T., Guan, X., Zheng, Q., Lu, K., Song, Y., & Zhang, W. (2009). Prototype Demonstration: Trojan Detection and Defense System. In *Consumer Communications and NETWORKING Conference* (pp. 1-2). IEEE. doi:10.1109/CCNC.2009.4785028

Mila Parkour. (2013). APT samples shared by Mila Parkour. Retrieved from https://www.mediafire.com/?a49l965nlayad#734479hwy1b97

Nuclear-EK-malware. (2014). A blog focuses on network traffic related to malware infections. Retrieved from http://www.malware-traffic-analysis.net/2014/index.html

Pallaprolu, S. C., Namayanja, J. M., Janeja, V. P., & Adithya, C. T. S. (2017). Label propagation in big data to detect remote access Trojans. In *IEEE International Conference on Big Data* (pp. 3539-3547). IEEE.

Pan, X. U., Sheng-li, L. I. U., Jing-hong, L. A. N., & Da, X. I. A. O. (2015). Trojan detection method based on analysis of multiple data flow. *Jisuanji Yingyong Yanjiu*, (03): 890–894.

Peng, G. J., Wang, T. G., Shao, Y. R., & Liu, M. L. (2012). *Technology and implementation to detect unknown trojan based on network flow characteristics*. Netinfo Security.

Pu, Y., Chen, X., Cui, X., Shi, J., Guo, L., & Qi, C. (2013). Data stolen trojan detection based on network behaviors. *Procedia Computer Science*, *17*, 828–835. doi:10.1016/j.procs.2013.05.106

Sanok, D. J. (2005). An analysis of how antivirus methodologies are utilized in protecting computers from malicious code. In *Conference on Information Security Curriculum Development* (pp. 142-144). ACM. doi:10.1145/1107622.1107655

Wei, L. I., Li-Hui, L. I., Jia, L. I., & Lin, S. W. (2015). *Characteristics analysis of traffic behavior of remote access trojan in three communication phases*. Netinfo Security.

Wu, N. Q., Qian, Y., & Chen, G. (2006). A Novel Approach to Trojan Horse Detection by Process Tracing. *IEEE International Conference on Networking, Sensing and Control* (pp.721-726). IEEE.

Xiang, B., Hao, Y. J., Zhang, Y., & Liu, H. Y. (2009). A Novel Anti-Trojan Approach using Behavioral Analysis. In *International Conference on Apperceiving Computing and Intelligence Analysis* (pp.311-314). IEEE.

Yamada, M., Morinaga, M., Unno, Y., & Torii, S. (2015). RAT-based malicious activities detection on enterprise internal networks. In *International Conference for Internet Technology and Secured Transactions* (pp.321-325). IEEE. doi:10.1109/ICITST.2015.7412113

Zhang, J., Tong, Y., & Qin, T. (2016). Traffic features extraction and clustering analysis for abnormal behavior detection. *International Conference on Intelligent Information Processing* (p. 25). ACM. doi:10.1145/3028842.3028867

Zhioua, S., Jabeur, A. B., Langar, M., & Ilahi, W. (2014). Detecting Malicious Sessions Through Traffic Fingerprinting Using Hidden Markov Models. *International Conference on Security and Privacy in Communication Systems* (pp.623-631). Springer International Publishing.

Xianda Wu is a graduate student studying at Beijing University of Technology. She works at Institute of Information Engineering, Chinese Academy of Sciences as an intern from September 2016 to now. Her main research is the network security which includes Trojan detection, malicious web page detection.

Wei Jiang, received his M.S. and Ph.D. degrees in information security from the Harbin Institute of Technology (HIT), China in 2006 and 2010 respectively. He is an associate professor and master's supervisors of Beijing University of Technology, China. His research interests are network and information security, cloud computing, social computing. Xiang Cui received his Ph.D. degree in Information Security from the Institute of Computing Technology, Chinese Academy of Sciences in 2012. Currently serves as a senior fellow at Guangzhou University Cyberspace Advanced Technology Research Institute. In 2003, after his graduate schooling, he worked in network security emergency response at CNCERT / CC. He directed participants in a series of major cyber-security incident and 863-917 platform construction. From 2007 to now, he studied cyberattack and defense technologies first at CAS Institute of Computing and later at Chinese Academy of Sciences. His research areas is network offensive and defensive techniques.

Chaoge Liu now serves as an assistant researcher at the Institute of Information Engineering, Chinese Academy of Sciences. His main research on web security, internet fraud, traceback, malicious code and so on. He published more than 10 articles on malicious code in international conferences and obtained 2 cooperation patents. As a core member, he participated in many projects such as the 863 Program and the National Natural Science Foundation of China.