Detection and Defense of Network Virus using Data Mining Technology

Zhijun Li (ju42781@163.com)
Hebei Normal University for Nationalities
Xuedong Jiang
Hebei Normal University for Nationalities

Research

Keywords: Data mining, network virus, application programming interface call, feature selection, virus detection

Posted Date: May 18th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-508107/v1

Detection and defense of network virus using data mining technology

Zhijun Li[1*], Xuedong Jiang[2]

[1]Integrated Business Department of Assets and Laboratory Management Center, Hebei Normal University for Nationalities, Chengde, Hebei 067000, China

[2]Information Center, Hebei Normal University for Nationalities, Chengde, Hebei 067000, China

Corresponding address: Assets and Laboratory Management Center, Hebei Normal University for Nationalities, Chengde, Hebei 067000, China

Email: ju42781@163.com

Abstract:

The spread of network viruses has posed a serious threat to the security of the network; therefore, it is necessary to detect and defend them effectively. This paper used Debug application programming interface (API) technology to obtain the features of API calls as viruses, filtered API calls according to information entropy, and finally used the support vector machine (SVM) model for virus detection. The experimental results showed that when the number of API was 1200, the algorithm had the best virus detection performance, with an average true positive rate (TPR) of 95.2%, a false positive rate (FPR) of 3.31%, and an overall accuracy of 95.42%; compared with the K-means algorithm and Naive Bayes algorithm, the SVM algorithm had the best performance. The results show that the proposed method is effective in virus detection and defense and can be further promoted and applied in practice.

Keywords: Data mining, network virus, application programming interface call, feature selection, virus detection

1. Introduction

The rapid development of the network has brought great conveniences to people's study, life, work, and entertainment (Gao et al., 2021), but at the same time, the security problem of the network has become increasingly prominent (Ma, 2017). Network virus refers to a group of codes that can destroy the function or data of computers and also has the ability of self-replication, i.e., it can spread quickly among the networks to affect the normal use of computers. Once the virus code is executed in computers, it will be inserted into various programs to multiply itself. If it is not handled in time, these viruses will spread to other uninfected computers through all platforms on the network, making more computers lose the ability to work normally and even causing the paralysis of the network system. In addition, the viruses have strong concealment, sometimes can not be detected by anti-virus software. Before triggering, the viruses will lurk in the program; once triggered, they will wantonly destroy the network. The increasingly rampant network virus severely challenges computer security (Zuo, 2018). The high-speed spread of information and the rampant virus bring a huge threat to people's information security. Therefore, how to realize the detection and defense of network viruses and reduce the damage of viruses to computers to ensure the normal work of the network has been widely concerned by researchers. With the emergence and development of data mining technology, it plays an increasingly important role in network virus defense. Binh et al. (2016) divided the control flow graph (CFG) extracted from a binary executable into specific subgraphs to extract malicious behaviors. The method could reduce the complexity of verification by incremental verification strategy. The experiment showed that the method could significantly improve the performance. Nguyen et al. (2015) proposed a maximum entropy-based method. During training, the features in the virus database were extracted to establish the upper model. During detection, the upper model was used to identify viruses based on the corresponding features of a

checked file. Saudi et al. (2015) designed an efficient Trojan horse detection model based on the sequential minimum optimization (SMO) algorithm and found through experiments that the method had a 98.2% correct rate and a 1.7% error rate. Jerbi et al. (2020) evolved a group of application programming interface (API) calls combined with the artificial malware patterns and based on the genetic algorithm and found through the test on different data sets that the method could significantly improve the detection rate. In this paper, based on the support vector machine (SVM), the detection and defense methods of network viruses were studied, and its performance was tested on the data set to verify the possibility of the method in practical application. This work makes some contributions to the further development of network security.

2. The SVM-based detection and defense method

2.1 Network virus feature screening

In the Windows system, PE file is the standard format of executable file (Belaoued and Mazouzi, 2018). Therefore, the virus is written according to the format of the PE file in order to spread better. In the operation process, Windows calls various service functions to realize some function, and the functions are called API functions. To read a file on disk, first of all, the Readfile function in Kernel32.dll is called; if the permission meets the requirement, the NtReadFile function in Ntdll.dll is called; then, CPU is switched to the operating system kernel mode through the KiSystemService function; finally, the NtReadFlie kernel function in ntoskrnl.exe is called. After file reading, the NtReadFile function in ntoskrnl.exe will return in the same way. This is an API call.

All programs on the Windows platform need API calls to implement functions, including the degree of viruses. Therefore, network viruses can be detected by capturing API calls, such as Windows debug API (Sigalov et al., 2017) and APIHOOK technology (Case et al., 2019). However, APIHOOK technology can only be implemented when every API prototype is known, and not every function can be effectively captured. Therefore, this study used Windows Debug API technology to obtain API calls.

Debug API technology can load a program or bind itself to the program to facilitate debugging. If there are debugging-related events, the debugger will be triggered, i.e., the monitoring process will be activated. The steps of debug API technology can be described as follows: (1) the monitor program is started after inputting samples; (1) the export table of the library file called by the debugging program is analyzed; (3) the breakpoint is set at the entry address of the function; (4) if an interrupt occurs at the breakpoint, the debug process information will be obtained; (5) the debugging process is cycled until the end, and a behavior report is obtained.

In the collected API calls, a large part of the content does not play a great role in virus detection but will cause information redundancy and increase the calculation of subsequent virus detection. Therefore, it is necessary to screen the collected features and select the features with a high differentiation degree and a high important degree. This paper used the method of information entropy. Regarding whether a program calls related API as information and whether a program is a

virus or not as an event, the contribution of information to an event was judged through the calculation of information entropy, so as to determine the importance of the feature.

The entropy of a random variable X is set as: $H(X)=-\sum_{i=1}^{k} p_i(X=v_i)\log_2 p_i(X=v_i)$, where X has k values and p_i represents the probability that the value of X is v_i. It is assumed that under the condition that a random variable Y has been known, the conditional entropy of X is: $H(X|Y)=-\sum_{j=1}^{m} p_j(Y=v_j)\times H(X|Y=v_j)$, where $H(X|Y)$ refers to the uncertainty of X. Then, under the condition that Y has been known, the information added value of X, i.e., the information grain, can be written as: $IG(X|Y)=H(X|Y)-H(X)$; the greater the value is, the larger the function of the API call is. Features with large information gains are reserved to establish a virus feature set. Every feature vector is represented by a boolean vector. If the feature appears in the program, it is assigned as 1; otherwise, it is assigned as 0. A program is set as B_i, and a feature attribute is set as A_i. The feature set can be represented by matrix $M_{A\times B}$:

$$M_{A\times B}=\begin{array}{c} \\ B_1 \\ B_2 \\ \mathrm{M} \\ B_m \end{array}\begin{array}{c} \begin{array}{cccc} A_1 & A_2 & \Lambda & A_n \end{array} \\ \begin{pmatrix} b_{11} & b_{12} & \Lambda & b_{1n} \\ b_{21} & b_{22} & \Lambda & b_{2n} \\ \mathrm{M} & \mathrm{M} & \Lambda & \mathrm{M} \\ b_{m1} & b_{m2} & \Lambda & b_{mn} \end{pmatrix} \end{array} \quad , (1)$$

where $b_{ij}=\begin{cases} 1, Program\ B_i\ has\ feature\ A_j \\ 0, Program\ B_i\ does\ not\ have\ feature\ A_j \end{cases}$.

2.2 The SVM-based detection mode

SVM is a method based on structural risk minimization (Shi et al., 2015). It uses quadratic programming to find the optimal solution, avoiding the local minimization problem of neural networks, and solves the dimension problem through the kernel function. Therefore, it has been widely used in many fields. Virus detection can be regarded as a classification problem. Therefore, this paper uses SVM to detect viruses.

It is assumed that a sample set is (x_i, y_i), where $i=1,2,\Lambda\ ,n$, $y\in\{+1,-1\}$, x_i represents the feature of the program, and y_i represents the class that the program belongs to. The corresponding classification surface can be written as $f(x)=wx+b$, i.e., satisfying $y_i(wx_i+b)\geq 0$, where w

stands for a weight and b stands for a bias. The class interval is $\frac{2}{\|w\|}$, the optimal classification surface can be written as: $\min \varphi(w)=\frac{\|w\|^2}{2}$, and the constraint condition is: $f(x) \geq 1, i=1,2,\Lambda\ ,n$. Then, the decision function is obtained:

$$g(x)=\operatorname{sgn}(wx_i+b). \quad (2)$$

In order to solve equation (2), Lagrange multiplier a is introduced to transform it into a dual problem:

$$Q(a)=\sum_{i=1}^{n} a_i-\frac{1}{2}\sum_{i,j=1}^{n} a_i a_j y_i y_j (x_i x_j). \quad (3)$$

The constraint condition is:

$$\begin{cases} \sum_{i=1}^{n} y_i a_i=0 \\ a_i \geq 0 \end{cases}. \quad (4)$$

Finally, the SVM model for virus detection can be written as:

$$f(x)=\operatorname{sgn}\left(\sum_{i=1}^{n} a_i^* y_i k(x_i \cdot x)+b^*\right)$$

where k is a kernel function. The common kernel functions are:

(1) linear kernel function: $K(x_i, x_j)=x_i^T x_j$;

(2) polynomial kernel function: $K(x_i, x_j)=(\gamma x_i^T x_j+r)^d, \gamma>0$;

(3) S-shaped kernel function: $K(x_i, x_j)=\tanh(\gamma x_i^T x_j+r)$;

(4) radial basis function (RBF) kernel function: $K(x_i, x_j)=\exp\left(-\gamma\|x_i-x_j\|^2\right), \gamma>0$.

γ, r, and d are all nuclear parameters.

In selecting the kernel function, because of the large number of samples and features required in virus detection, the RBF kernel function can avoid the dimension disaster better. Therefore, this paper chose the RBF kernel function to build the SVM model.

3. Experimental analysis

Virus samples were collected from some well-known forums and laboratories, and normal samples were also selected from the Windows system. Finally, 1256 samples were obtained, including 692 virus samples and 564 normal samples. Then, features were extracted by debug API technology, and the obtained features were stored in the MySQL database.

The purpose of the experiment was to analyze the virus detection and defense ability of the model designed in this paper. The judgment result was “yes” or “no”, as shown in Table 1.

Table 1 Classification results

Results	Normal	Virus
Normal file	TP	FN
Virus files	FP	TN

The evaluation indexes of the model included:

true-positive rate: $TPR = \frac{TP}{TP + FN}$;

false-positive rate: $FPR = \frac{FP}{FP + TN}$;

total accuracy: $Total\_Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$.

Firstly, the influence of the number of features on the performance of virus detection was analyzed. After extracting API calls by using debug API technology, a total of 2167 API calls were obtained. Through feature screening, different numbers of APIs were selected for virus detection. The overall accuracy of the model is shown in Table 2.

Table 2 Changes of overall accuracy

API quantity/n	Overall accuracy/%
100	80.79
300	86.78
600	92.34
900	93.62

1200	95.27
1500	91.88
1800	91.36
2100	90.27

It was seen from Table 2 that with the increase of the number of APIs, the overall accuracy of the model rose rapidly, from about 80% to about 90%. When the number of APIs reached 1200, the overall accuracy was the highest, 95.27%. Then, with the continuous growth of the number of APIs, the accuracy of the algorithm began to decline. When the number of APIs reached 2100, the overall accuracy of the algorithm was 90.27%, which decreased by 5%. Therefore, the number of selected APIs was set as 1200.

The method of ten-fold cross-validation was adopted to determine the performance of the SVM model, as shown in Table 3.

Table 3 Virus detection results of the SVM model

Verification times	TPR/%	FPR/%	Overall accuracy/%
1	95.64	3.27	96.12
2	94.87	4.12	95.65
3	96.33	2.36	96.11
4	94.39	3.25	94.36
5	95.85	2.12	95.18
6	96.21	3.51	94.82
7	92.58	3.64	95.88
8	93.69	3.98	96.09
9	97.08	4.26	94.39
10	95.39	2.61	95.64
Average	95.20	3.31	95.42

It was seen from Table 3 that the average TPR and FPR of the SVM model were 95.2% and 3.31% respectively, and the overall accuracy rate reached 95.42%, which indicated that the feature screening and detection model designed in this paper had a high accuracy rate in the detection and defense of network viruses. In order to further prove the effectiveness of the model, it was compared with the K-means algorithm and the Naive Bayes algorithm. The experimental results are shown in Figure 1.

It was seen from Figure 1 that the TPR values of the three algorithms were 93.22%, 94.26%, and 95.2%, respectively, i.e., the TPR of the SVM model was 1.98% and 0.94% higher than that of the former two algorithms; the FPR values of the three algorithms was 6.32%, 4.67%, and 95.2%, respectively, i.e., the FPR of the SVM model was 3.01% and 1.36% lower than that of the former two algorithms. The overall accuracy of the three algorithms were 88.67%, 92.16%, and 95.42%, respectively. The SVM model had the highest accuracy, which was 6.75% and 3.26% higher than the former two algorithms, which verified the reliability of the SVM model.

4. Discussion

The emergence of more and more new and unknown viruses not only will have a huge impact on society but also may lead to irreparable economic losses (Wang, 2017); therefore, virus detection and defense has become a key and difficult problem in the field of network security (Tang et al., 2020), and more new and effective methods are urgently needed. At present, the commonly used methods include the behavior detection method (Choi et al., 2019), the heuristic scanning method (Zakeri et al., 2015), the intelligent detection method, etc. The intelligent detection method refers to applying intelligent algorithms such as data mining and machine learning to realize the detection and defense of viruses. Many data mining algorithms have been successfully applied in virus detection, such as decision trees and neural networks. It can be found that data mining has great potential and prospects in virus detection and defense.

In this study, API calls were extracted by debug API technology as virus features and screened based on information entropy, and the SVM model was used as a detection model to study the network virus detection. First of all, the number of extracted features had an impact on the virus detection performance. When the number of features was too large, the redundant information contained will not be conducive to virus detection. According to Table 2, when the number of APIs used was 1200, the algorithm had the best performance. In the ten-fold cross test, the SVM model showed 95.2% TPR, 3.31% FPR, and 95.42% overall accuracy, indicating that the SVM model had an excellent performance in virus detection and could effectively distinguish normal and virus programs. Finally, compared with the other algorithms, the SVM model had good reliability in virus detection.

In this paper, although some results have been obtained from the research of network virus detection and defense methods, there are still some shortcomings. In future research, the author will:

(1) analyze more data mining methods;

(2) further improve the performance of virus detection;

(3) further subdivide the network viruses and compare Trojan horse viruses with worm viruses.

5. Conclusion

Based on the SVM model, this paper analyzed the method of network virus detection and defense and collected virus programs and normal programs as data sets for experiments. The results showed that the performance was the best when 1200 API calls were selected for virus detection. The average TPR of the SVM model was 95.2%, the average FPR was 3.31%, and the overall accuracy was 95.42%. Compared with the K-means algorithm and Bayesian algorithm, the SVM model had a better performance; therefore, the SVM model can be further promoted and applied in practice.

List of abbreviations

API: application programming interface

SVM: support vector machine (SVM)

FPR: false positive rate

TPR: true positive rate

CFG: control flow graph

SMO: sequential minimum optimization

Declaration Statements

Availability of data and material

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Competing interests

The authors declare that they have no competing interests.

Funding

Not applicable.

Authors' contributions

ZJL conceived the idea. ZJL and XDJ designed the experiment and carried out the experiment. XDJ analyzed the data. XDJ wrote the first draft. ZJL revised the manuscript.

Acknowledgements

Not applicable.

References

A Case, MM Jalalzai, M Firoz-Ul-Amin, RD Maggio, A Ali-Gombe, MX Sun, GG Richard, HookTracer: A System for Automated and Accessible API Hooks Analysis. Digit. Invest. 29, S104-S112 (2019). doi:10.1016/j.diin.2019.04.011.

C Choi, C Esposito, M Lee, J Choi, Metamorphic malicious code behavior detection using probabilistic inference methods. Cogn. Syst. Res. 56(AUG.):142-150 (2019). doi:10.1016/j.cogsys.2019.03.007.

C Zuo, Defense of Computer Network Viruses Based on Data Mining Technology. Int. J. Netw. Secur. 20(4):805-810 (2018). doi:18163548-201807-201807180001-201807180001-805-810.

DA Sigalov, AV Razdobarov, AA Petukhov, Detecting DOM-based XSS vulnerabilities using debug API of the modern web-browser. Prikl. Diskr. Mat. 35, 63 – 75 (2017). doi:10.17223/20710410/35/6. doi:10.17223/20710410/35/6.

J Shi, WJ Lee, Y Liu, Y Yang, P Wang, Forecasting Power Output of Photovoltaic Systems Based on Weather Classification and Support Vector Machines. IEEE T. Ind. Appl. 48(3):1064-1069 (2015). doi:10.1109/TIA.2012.2190816.

M Belaoued, S Mazouzi, A Real-Time PE-Malware Detection System Based on CHI-Square Test and PE-File Features. IFIP Adv. Inform. Commun. Tech., 456, 416-425 (2018). doi:10.1007/978-3-319-19578-0_34.

M Jerbi, ZC Dagdia, S Bechikh, M Makhlouf, LB Said, On the use of artificial malicious patterns for android malware detection. Comput. Secur. 92(May):101743.1-101743.22 (2020). doi:10.1016/j.cose.2020.101743.

M Zakeri, FF Daneshgar, M Abbaspour, A static heuristic approach to detecting malware targets. Secur. Commun. Netw.8(17):3015-3027 (2015). doi:10.1002/sec.1228.

MM Saudi, AM Abuzaid, BM Taib, ZH Abdullah, Designing a new model for Trojan horse detection using sequential minimal optimization. Lect. Notes Electr. Eng. 315, 739-746 (2015). doi:10.1007/978-3-319-07674-4_69.

NT Binh, QT Tho, HM Ngoc, NM Hai, Incremental verification of ω-regions on binary control flow graph for computer virus detection. 2016 3rd National Foundation for Science and Technology Development Conf. on Information and Computer Science (NICS), 68-73 (2016). doi:10.1109/NICS.2016.7725670.

NT Nguyen, PV Huong, BC Le, DT Le, TH Van Le, A New Method of Virus Detection Based on Maximum Entropy Model. Adv. Intell. Syst. Comput. 358, 151-161 (2015). doi:10.1007/978-3-319-17996-4_14.

W Tang, YJ Liu, YL Chen, YX Yang, XX Niu, SLBRS: Network Virus Propagation Model based on Safety Entropy. Appl. Soft Comput. 97, 106784 (2020). doi:10.1016/j.asoc.2020.106784.

X Gao, L Liu, X Zhu, Research on the Main Threat and Prevention Technology of Computer Network Security. IOP Conf. Ser. Earth Environ. Sci. 632(5):052065 (7pp) (2021). doi:10.1088/1755-1315/632/5/052065.

XJ Ma, Research and Implementation of Computer Data Security Management System. Proc. Eng. 174(Complete):1371-1379 (2017). doi:10.1016/j.proeng.2017.01.290.

Y Wang, Food Information Management and Security Strategy of Computer Network. Adv. J. Food Sci. Tech.11(12):792-794 (2017). doi:10.19026/ajfst.11.2793.

Figure 1 Performance comparison of different algorithms

Figures

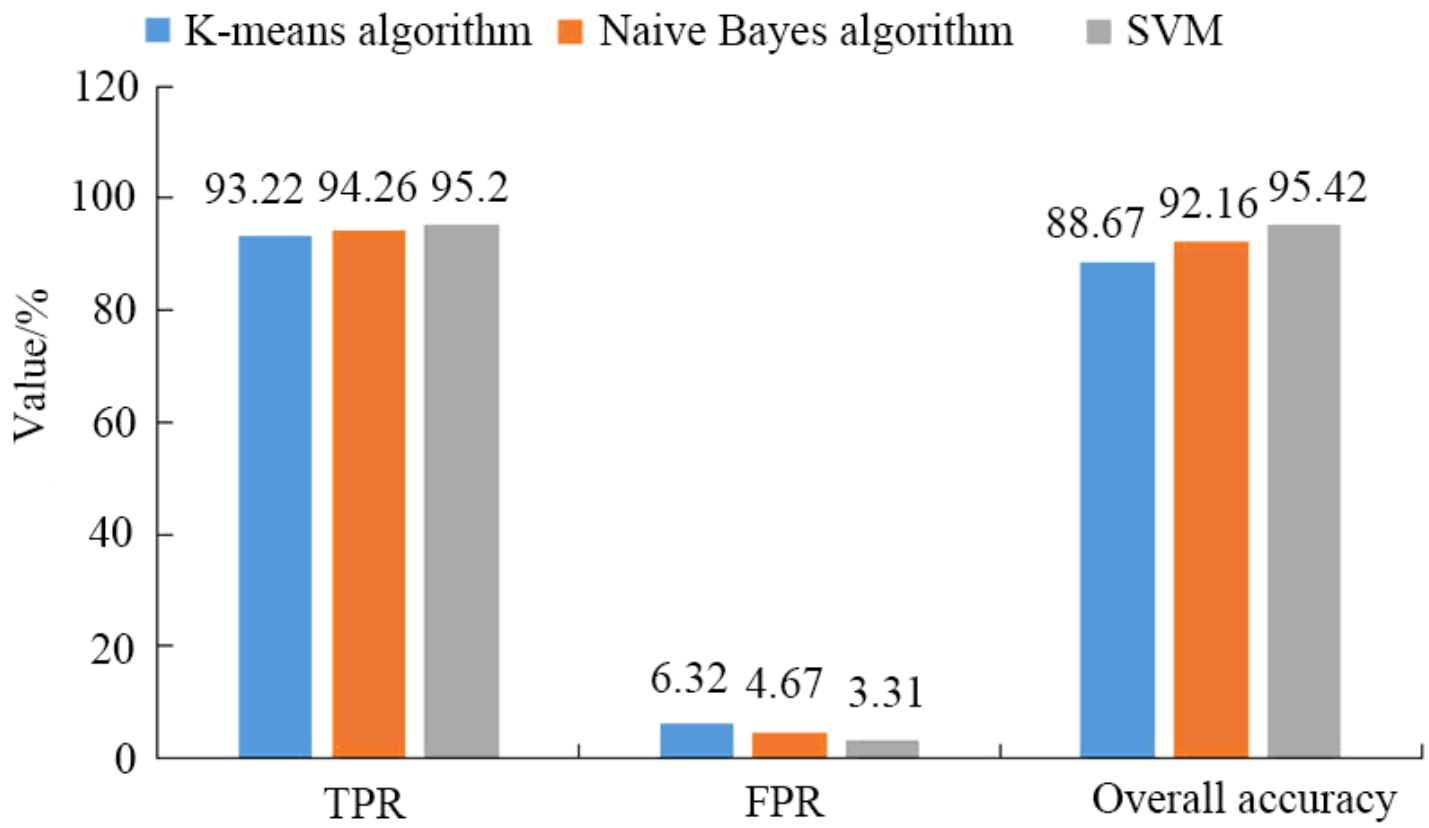

Figure 1

Performance comparison of different algorithms