IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 15, 2019, accepted October 4, 2019, date of publication October 10, 2019, date of current version October 24, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2946482

A Novel Solutions for Malicious Code Detection and Family Clustering Based on Machine Learning

HANGFENG YANG[1], SHUDONG LI[1], XIAOBO WU[2], HUI LU[1], AND WEIHONG HAN[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

Corresponding authors: Shudong Li (lishudong@gzhu.edu.cn), Hui Lu (luhui@gzhu.edu.cn), and Weihong Han (hanweihong@gzhu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61672020, Grant U1803263, Grant 61972106, Grant U1636215, and Grant 61972105, in part by the National Key Research and Development Program of China under Grant 2019QY1406, and in part by the Key Research and Development Program of Guangdong Province under Grant 2019B010136003.

ABSTRACT Malware has become a major threat to cyberspace security, not only because of the increasing complexity of malware itself, but also because of the continuously created and produced malicious code. In this paper, we propose two novel methods to solve the malware identification problem. One is to solve to malware classification. Different from traditional machine learning, our method introduces the ensemble models to solve the malware classification problem. The other is to solve malware family clustering. Different from the classic malware family clustering algorithm, our method introduces the t-SNE algorithm to visualize the feature data and then determines the number of malware families. The two proposed novel methods have been extensively tested on a large number of real-world malware samples. The results show that the first one is far superior to the existed individual models and the second one has a good adaptation ability. Our methods can be used for malicious code classification and family clustering, also with higher accuracy.

INDEX TERMS Malware, ensemble model, malware classification, family clustering, t-SNE.

I. INTRODUCTION

Malware software refers to software programs that purposely implement the harmful intentions of an attacker [1], [2], especially on the IOT [3], [4] and network security [5], [6]. The intent of such malware includes disrupting the normal operation of the system, attempting to acquire resources of the computer system or network, obtaining such resources without the user's permission, and stealing private sensitive information [7], [8]. Therefore, malware software poses a huge threat to the security of the host, the security of the network and privacy [9], [10], and the smart devices [11], [12]. According to the literature, in the past few quarters, the total number of malware software samples has increased by 34% to more than 774 million samples. At the same time, with the rapid development of the software industry, malware software has also developed extensively, and it has become extremely complicated.

With the dramatic increase in the amount of malware, substantial research has focused on malware detection technology. In general, malware detection techniques can be divided into two categories: static malware analysis and dynamic malware analysis. On one hand, static malware analysis is more efficient and safer than other methods because it can be analyzed without executing code. But it is also susceptible to evasion techniques. On the other hand, dynamic malware analysis is better able to detect unknown malware. However, because the analysis process is performed in a virtual environment, it results in more and more consumption of system resources.

The analysis and detection of malware have always been a challenge for security experts. Traditional attempts have focused on static and dynamic analysis. Then, the rapid growth and evolution of malware has forced researchers to introduce new analysis and detection solutions. Machine learning is one of the innovative technologies applied in this direction [14]. Substantial amounts of works have focused on building the frameworks for analysis [15], [16], obtaining static features [17], [18], and classifying malware families [19]–[21]. The previous experiments show that the text classification method is very effective in improving the detection accuracy of fuzzy samples. For comparison, the various ML algorithms, including naive Bayes, random forest, and support vector machine (SVM), have been

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaojiang Du.

utilized to detect sequences of malicious application programming interface (API) calls [18]. By using n-gram instead of byte sequences, the performance of naive Bayes, decision tree, and SVM in malware detection are compared [22]. As methods of detecting malware by using data mining technology [23], it was first proposed to use three different types of static features: PE headers, string sequences and byte sequences. In alternative approaches to exploring and utilizing sample visual features [24], most studies have considered that malware can be clustered based on families or similarity [25]–[27]. Also, Artificial neural networks [28], [29] have been used for malware detection. Recently, novel ideas have been applied to malware detection [30]. For example, the image processing technology is used to detect malicious software [31].

In terms of malware detection, preliminary works have achieved sufficient performance. However, in general, firstly, whether it is static malware analysis or dynamic malware analysis, most methods only used a single algorithm to identify malware. Meanwhile, in terms of both accuracy and other relevant evaluation metrics, the previous malware identification methods still have significant limitations. Secondly, the identification of malware does not only include classification, clustering is also a very effective technique which can discover unknown malware. Therefore, one of the challenges in the research on malware families clustering is the determination of the number of malware families.

In order to solve the above-mentioned problems, this paper adopts the idea of multi-model ensemble to identify malware and introduces t-SNE algorithm to determinate the number of malware families. The contribution of this work is as follows.

(a) For malware classification, we introduce the ensemble model to overcome the deficiency of single training model. As a result, the ensemble model is far superior to the existed individual models. First, the feature engineering is performed on the data set A. Then, the feature matrix generated by feature engineering is used to independently train multiple classifiers. The recognition results of the classifiers are integrated by some ensemble strategy, and the integrated result output is used as the final recognition result.

(b) For malware family clustering, we use the data visualization method to determine the number of malware families and find that the method has a good adaptation ability. Similarly, the data set needs to be characterized first. Then, we use a visualization algorithm t-SNE to visualize the feature matrix generated by feature engineering. Through the visualization process, the number of malware families can be estimated. Finally, some clustering algorithms are used to perform the clustering of malware families.

The reminder of this paper is organized as follows. In Section 2, we mainly give the description of two problems that this paper focuses on. In section 3, we describe the dataset used in this paper. In section 4, we give the model establishment of this work, including the feature engineering and the two models for two problems above. In section 5, the experimental results on dataset is analyzed. The section 6 gives the conclusion and our future work.

II. PROBLEM DESCRIPTION

In this paper, we try to solve the following two problems in malware detection by using machine learning. Fig.1 is the excerpt from the xml file output after an exe file is run through the sandbox.

```
- <action status_value="0" ret_value="0" err_code="0" call_time="07:11:41.006" call_pid="396" call_name="21029235.exe"
  api_name="TryToAnalyze">
      <apiArg_list count="0"/>
    - <exInfo_list count="1">
        <exInfo value="C:\program\21029235.exe"/>
      </exInfo_list>
  </action>
- <action status_value="0" ret_value="0" err_code="0" call_time="07:11:41.019" call_pid="396" call_name="21029235.exe"
  api_name="LoadLibraryExW">
    - <apiArg_list count="3">
        <apiArg value="0"/>
        <apiArg value="0"/>
        <apiArg value="0"/>
      </apiArg_list>
    - <exInfo_list count="3">
        <exInfo value="C:\WINDOWS\system32\advapi32.dll"/>
        <exInfo value="00077da0000"/>
        <exInfo value="000a9000"/>
      </exInfo_list>
  </action>
- <action status_value="0" ret_value="0" err_code="0" call_time="07:11:41.021" call_pid="396" call_name="21029235.exe"
  api_name="LoadLibraryExW">
    - <apiArg_list count="3">
        <apiArg value="0"/>
        <apiArg value="0"/>
        <apiArg value="0"/>
      </apiArg_list>
    - <exInfo_list count="3">
        <exInfo value="C:\WINDOWS\system32\rpcrt4.dll"/>
        <exInfo value="00077e50000"/>
        <exInfo value="00092000"/>
      </exInfo_list>
  </action>
```


FIGURE 1. Excerpt from the xml file output after an exe file is run through the sandbox.

(a) Problem One: with the dynamic behaviors of a large number of PC malware viruses, the first main important problem is how to extract malware features of malware from documents and train classifier to classify a virus based on these features. In this paper, the documents are the xml files outputted by an exe file after the sandbox is run is used to determine whether the file is malware. In order to solve this problem, the key two parts are the feature engineering and the training classifier, and here we used the ensemble strategy to construct the classifier with high accuracy.

(b) Problem Two: according to the dynamic behaviors of a large number of PC malware viruses, the second important problem is how to quickly identify and accurately detect the same family's malware mutation, namely the malware family clustering, where the corresponding clustering algorithm is designed to judge the family id of the xml file output by an exe file after passing through the sandbox operation. And the key point for this problem is the determination of the number of cluster centers k.

III. DATASET DESCRIPTION

A. DATASET USED IN MALWARE CLASSIFICATION

The data set for malware classification contains a total of 30,000 sample data, including 10,000 black samples and 20,000 white samples. The total sample size is about 20G [32]. The file type is XML file. It includes the execution action sequence of a certain software, such as return value, call time, call process number, call API name and other information.

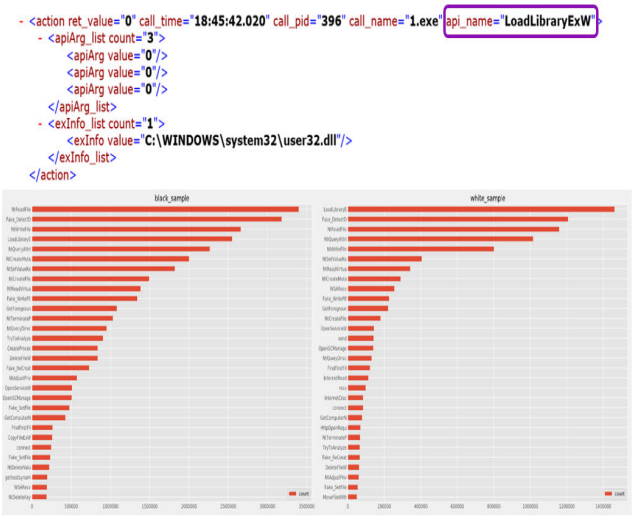

FIGURE 2. Exploring the feature distribution corresponding to the api_name field in the black and white samples.

FIGURE 3. Exploring the feature distribution corresponding to the call_pid field in the black and white samples.

B. DATASET USED IN MALWARE FAMILY CLUSTERING

The data set for family clustering has a total of 60,000 samples, the data set size is about 50G, and the file type is XML file [32]. The data are also from the DataCon big data security analysis competition held by qianxin.

IV. MODEL ESTABLISHMENT

A. FEATURE ENGINEERING

In the data exploration stage, by statistically analyzing the proportion of the features corresponding to each field in the black and white samples, it can be found that the features corresponding to the relevant fields have significant differences in the black and white samples regardless of the type of features or the number of features. In this paper, we statistically analyze the distribution of features corresponding to four fields: api_name, call_pid, ret_value, and exInfo.

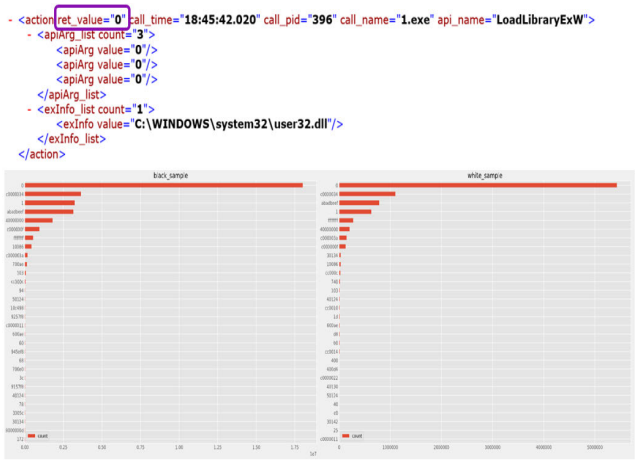

FIGURE 4. Exploring the feature distribution corresponding to the ret_value field in the black and white samples.

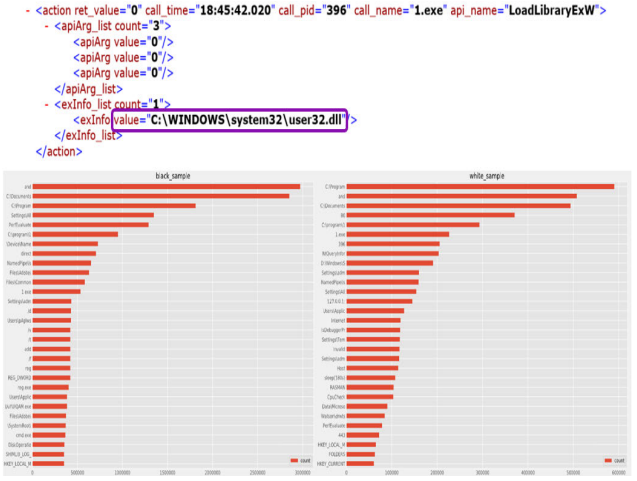

FIGURE 5. Exploring the feature distribution corresponding to the exInfo field in the black and white samples.

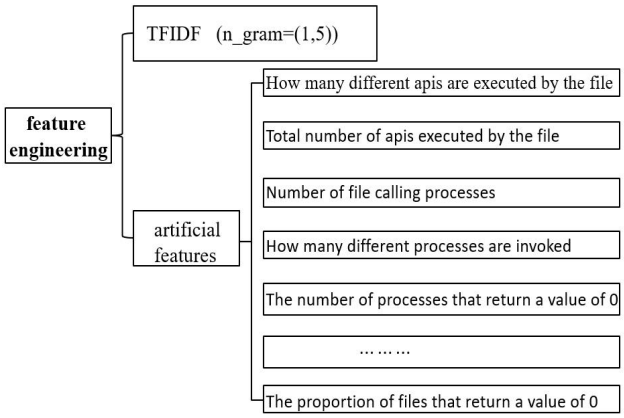

FIGURE 6. Feature engineering diagram.

The api_name field represents the API name that is called when the program executes. When statistical analysis is performed on the features corresponding to the api_name field,

it is found that, although there is no significant difference in the feature types corresponding to the api_name field in the black and white samples, there is a very large difference in the number of different features of the black and white samples.

The call_pid field represents the process number that is called when the program executes. In the statistical analysis of the features corresponding to the call_pid field, it is found that there is a very large difference in the number and types of different features of the black and white samples. An interesting phenomenon is also found. As long as the program calls the process with the pid of 1244, the program must be malware software; after the subsequent subdivision, it is found that when the program calls the process with pid 1396, if the number of occurrences of the feature is 254, 256, 257, 258, 259, or 262, it must be a normal program.

The ret_value field represents the return value that is called when the program executes. In the statistical analysis of the features corresponding to the ret_value field, there is also a significant difference in the number of features corresponding to the ret_value field in the black and white samples.

The exInfo field represents the extra program call information during the execution of the original program. Subjectively, the corresponding feature of the field is definitely helpful in distinguishing between the malware software and the normal program. When the feature statistics analysis of the exInfo field is performed, it is found that there are very large differences in the number and types of different features in the black and white samples. It is also found that if the number of occurrences of the feature reg is greater than or equal to 2 or the number of occurrences of the feature add is greater than or equal to 6, the program must be malware.

In summary, the following results are obtained for the exploration results between the feature distributions corresponding to each field.

Our feature engineering mainly consists of two parts. First, TF-IDF algorithm and n-gram algorithm are used to process all feature fields, TF-IDF algorithm is used to obtain the frequency and difference characteristics of feature fields, while n-gram algorithm can be used to obtain the context relationship between the fields. Second, artificial features can introduce the experience information of relevant fields, which can greatly enrich the diversity of features and improve the performance of the model.

B. BUILDING A MODEL FOR PROBLEM ONE

The malware software is classified by multi-model ensemble. The ensemble strategy adopts stacking that is shown in Fig. 7. The input layer uses the TF-IDF algorithm and Word2Vec to extract features and then integrates the features of the two algorithms; it then inputs them into the Layer 1 layer. Layer 1 includes 10 base classifiers, including MultinomiaNB, BernoulliNB, KNN, GBDT, SVM, and XGBoost. The selection strategy of the model is determined according to the purpose of the problem. Because the problem can ultimately be expressed as a dichotomy problem, 10 generic classification models are selected as the base model in the

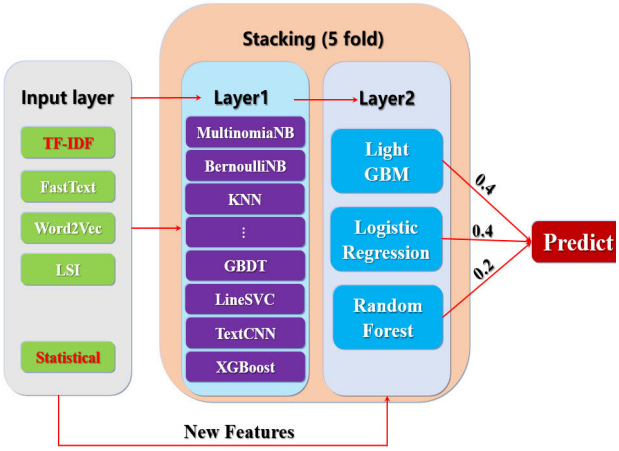

FIGURE 7. Malware software classification model architecture diagram.

first layer. After stacking, the output of Layer 1 is used as the input of Layer 2, and Layer 2 includes base classifiers such as LightGBM, Logistic Regression, and Random Forest. The base classifier output is multiplied by three adjustable coefficients $\alpha, \beta, \gamma \in (0, 1)$, here we suppose $\alpha = 0.4, \beta = 0.4, \gamma = 0.2$. Then, the weighted integration produces the final prediction.

The stacking ensemble strategy procedure is as follows in Fig.8.

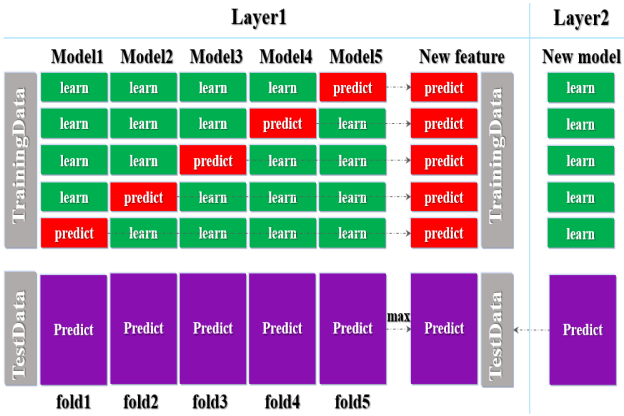

FIGURE 8. Stacking flowchat.

(a) Select the base model.
(b) Divide the training set into five non-intersecting sets, recorded as train 1, train 2, …, train 5.
(c) Starting from train 1 as a prediction set, use train 2 to train 5 to build the model; then, predict the training set and retain the result. Next, use train 2 as the other four training sets of the prediction set to build the model. Then, predict each time from train 1 to train 5.
(d) The predicted result is filled in accordance with the position of train 1 to train 5, and a stacking conversion is performed on the first base model of the entire training set.

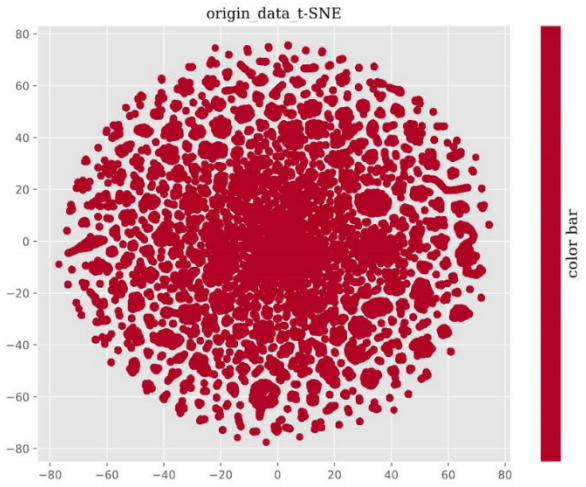

FIGURE 9. Raw sample data using t-SNE visualization results.

(e) In the above process of establishing five models, each model separately predicts the test set and ultimately retains the five columns of results. Then, it takes the row and the number of the five columns as the first base model to the test set a stacking transformation.
(f) Select the m base model and execute step 2.
(g) There are several base models that generate several new features for the entire training set. Similarly, several new features are generated for the test set.

In the step (e), we improved the traditional Stacking algorithm process, via replacing the average with the most frequent number in Test Data is to keep the predicted output of the test set consistent with the data type of the predicted output of the training set.

C. BUILDING A MODEL FOR PROBLEM TWO

The malware software family clustering problem is set in the feature engineering module. In the process of extracting features from TF-IDF, the maximum feature number is set to 100000, and then, the 200-dimensional features extracted from the Word2Vec algorithm [19] and the 20 human characteristics of the statistics are collected. Finally, all integrated features are reduced in dimension by the SVD algorithm so that the data matrix feature dimension can be reduced to 3000 dimensions.

This paper proposes a very novel method to cluster the malware code family. t-SNE is a nonlinear dimensionality reduction algorithm for mining high-dimensional data. It can map multidimensional data into two-dimensional or three-dimensional space. t-SNE algorithm can map high-dimensional data to low-dimensional space, which can keep a great degree of similarity between the original feature space and the feature space after dimension reduction in the whole process. Therefore, the number of clusters k can be observed by visualizing the space after dimension reduction.

Therefore, t-SNE is very suitable for visualization of high-dimensional data. First, the sample size is greatly reduced by randomly sampling the sample data, and then, the sample data are visualized by using the t-SNE algorithm [33], [34] so that the number of malware code families can be determined. For the k-means algorithm, the most difficult step is the determination of the number of cluster centers k. Therefore, in this paper, we use the t-SNE algorithm for visualizing the k-means algorithm to continuously adjust the parameter k to perform the comparison and achieve a good clustering performance. The whole framework is shown in Fig. 10.

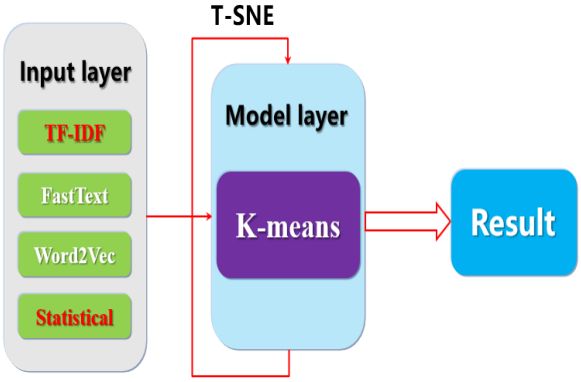

FIGURE 10. Malware software family clustering architecture diagram.

V. EXPERIMENTAL RESULT

A. RESULT ANALYSIS FOR PROBLEM ONE

The performance of the classification model is measured by three evaluation indicators: Accuracy, Recall, Precision and F1 score. To better describe the problem, the definitions of true positive (TP), false positive (FP), true negative (TN), and false negative (FN) are introduced, and the definition of accuracy is given, that is, the rate at which the correct result is predicted. The expression is as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

The precision concerns the prediction result and means the probability that it is actually a positive sample in all samples that are positively predicted. The expression is as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

The recall concerns the original sample and means the probability of being predicted as a positive sample in a sample that is actually positive. The expression is as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

The F1 score considers both the accuracy rate and the recall rate such that both are maximized while also achieving a balance, and the F1 Score expression is

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Result analysis for PROBLEM ONE:

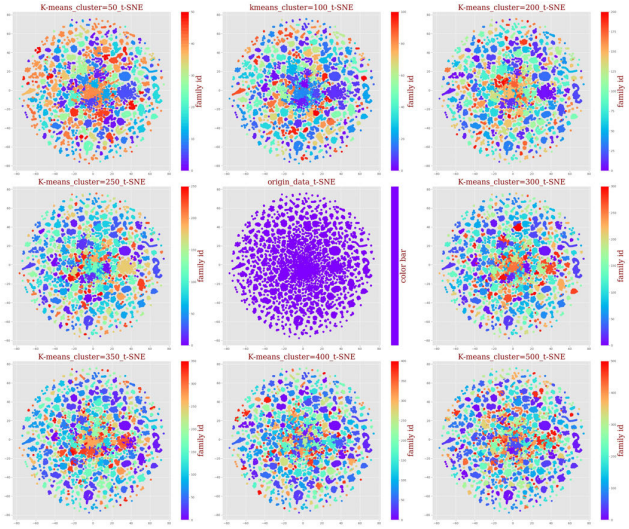

FIGURE 11. The t-SNE visual clustering performance for different k values.

(a) After feature integration of different feature extraction algorithms, the recognition performance of a single algorithm on malware software is also good.
(b) Multi-model ensemble is better for malware software recognition than single-model recognition. The multi-model ensemble for problem one can get very high accuracy for classification (see the data in Table 1).

TABLE 1. Results of the malware classification model.

Model	Accurcy	Recall	F1 Score
Random Forest	0.9800	0.9800	0.9800
XGBoost	0.9800	0.9700	0.9800
BPNN	0.9635	0.9635	0.9635
Decission Tree	0.9500	0.9500	0.9500
LR	0.9600	0.9700	0.9400
Naïve Bayes 1	0.9400	0.9500	0.9400
Naïve Bayes 2	0.9484	0.9484	0.9484
GBDT	0.9700	0.9600	0.9700
Bagging	0.9700	0.9600	0.9700
AdaBoost	0.9521	0.9521	0.9521
SVM	0.9700	0.9500	0.9600
Our model 1	**0.9839**	**0.9839**	**0.9839**
Our model 2	**0.9967**	**0.9967**	**0.9967**

(c) Different integration strategies have a greater impact on the recognition of malware software.

This part of the experiment has never theoretically discussed that how many base models are suitable for each layer and what features in different models are applicable to. If the above two difficulties can be solved, the training speed and accuracy of the model will be greatly improved.

TABLE 2. Joint value corresponding to different k values.

The value of K	Joint value
9	3.69
10	4.08
50	20.76
100	34.78
250	37.84
300	46.18
450	46.26
600	44.46

B. RESULT ANALYSIS FOR PROBLEM TWO

PROBLEM TWO uses the Joint Value to measure the performance of the clustering model, where the calculating formula for the Joint Value is as follows (5), shown at the bottom of this page, where N is the number of test samples. $F_{k\in F}$ is the predicted Family_id label set for the samples, $F_{k\in F} = \{k_1, k_2, \ldots\}$, where the element k_i represents that the k_ith sample belongs to the predicted Family_id $F_{k\in F}$. $FTrue_{k\in FTrue}$ is the real Family_id label set, where the elements are the all samples that belong to the real Family_id.

For example, for one simple data set with 6 samples, the predicted labels are $F_1 = \{5, 6\}$, $F_2 = \{1,2\}$, $F_3 = \{3,4\}$, and the real labels have 2 clusters FTure1 = {1,2,3,4}, FTure2 = {5,6}. Then, for sample 1, it belongs to F_2 in the predicted labels (namely the submitted answer) and FTrue1 in the real labels (namely the standard answer), respectively. So, according to the formula (5), the calculating score for sample 1 is:

$$\frac{\text{Count}\,(\text{F}_2 \cap \text{FTrue}_1)}{Count\,(\text{F}_2) + Count\,(\text{FTrue}_1) - \text{Count}\,(\text{F}_2 \cap \text{FTrue}_1)} = \frac{2}{2+4-2} = 0.5 \tag{6}$$

Then, we can get the Joint Value by calculating the average value of the scores for all samples. Now, we can see that the higher the joint value is, the better the clustering algorithm is.

Result analysis for PROBLEM TWO:

(a) After feature integration of different feature extraction algorithms, the same malware software has a higher degree of aggregation.
(b) After visualizing the data using the t-SNE algorithm, the k-value of the k-means algorithm is well determined, and there is no need to blindly select the k value, which greatly reduces the time consumption.
(c) For the clustering problem of malware software in oversized categories, the proposed scheme can achieve better results.

t-SNE algorithm was used to perform dimensionality reduction visualization on the original high-dimensional data,

$$\text{JointValue} = \frac{1}{\text{N}} \sum_{\text{k}=1}^{\text{N}} \frac{\text{Count}\,(F_{k\in F} \cap FTrue_{\text{k}\in\text{FTrue}})}{Count\,(F_{k\in F}) + Count\,(FTrue_{k\in FTrue}) - \text{Count}\,(F_{k\in F} \cap FTrue_{\text{k}\in\text{FTrue}})} \tag{5}$$

and it was concluded that the number of clustering clusters may cause some trouble when the number of clustering clusters is very large. The more the number of clusters is, the closer the visualization results will be, which makes it difficult to distinguish different clusters.

VI. CONCLUSION

In this paper, we proposed two novel solutions for identifying malware code. The first solution for problem one is to solve the malware classification by using an integrated learning method. The second solution for problem two uses the t-SNE algorithm to visualize the feature matrix to determine the number of malware families, and then adopts the classic k-means clustering algorithm to perform the malware family clustering. The experimental results show that, the proposed solutions have achieved good results on a malware sample set in a real-world environment, especially the multi-model ensemble for problem one can get very high accuracy for classification. On the other hand, the idea of the visual character matrix proposed in this paper for determining the number of malware families is very novel and has satisfied expectation.

In the future, the new methods proposed in this paper could be generalized to detect the malicious code and family clustering for IOT devices and wireless sensor networks [35]–[37]. On the other hand, in terms of feature engineering, our future research will focus on the semantic information of each field sequence, as it was not reflected in this paper. Also, in terms of model integration, this paper does not explore that which model used in the integration process will improve the classification performance. Thus, we will explore this point in the future.

AUTHOR CONTRIBUTIONS

Conceptualization: Shudong Li and Weihong Han. Methodology and formal analysis: Hangfeng Yang. Writing, review and editing: Shudong Li, Xiaobo Wu and Hui Lu. Funding acquisition: Shudong Li and Weihong Han.

REFERENCES

[1] U. Bayer, A. Moser, C. Kruegel, and E. Kirda, "Dynamic analysis of malicious code," *J. Comput. Virol.*, vol. 2, no. 1, pp. 67–77, 2006.

[2] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A distributed deep learning system for Web attack detection on edge devices," *IEEE Trans. Ind. Informat.*, to be published. doi: 10.1109/TII.2019.2938778.

[3] G. Li, J. He, S. Peng, W. Jia, C. Wang, J. Niu, and S. Yu, "Energy efficient data collection in large-scale Internet of Things via computation offloading," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4176–4187, Jun. 2019.

[4] X. Zhang, S. Yu, J. Zhang, and Z. Xu, "Forwarding SDN-based Internet of Things," *IEEE Internet of Things J.*, vol. 6, no. 2, pp. 3373–3385, Apr. 2019.

[5] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway, "A survey of key management schemes in wireless sensor networks," *Comput. Commun.*, vol. 30, nos. 11–12, pp. 2314–2341, Sep. 2007.

[6] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "An effective key management scheme for heterogeneous sensor networks," *Ad Hoc Netw.*, vol. 5, no. 1, pp. 24–34, 2007.

[7] Y. Xiao, X. Du, J. Zhang, and S. Guizani, "Internet protocol television (IPTV): The killer application for the next generation Internet," *IEEE Commun. Mag.*, vol.45, no. 11, pp. 126–134, Nov. 2007.

[8] X. Du and H. H. Chen, "Security in wireless sensor networks," *IEEE Wireless Commun. Mag.*, vol. 15, no. 4, pp. 60–66, Aug. 2008.

[9] Y. Ye, T. Li, D. Adjeroh, and S. S. Iyengar, "A survey on malware detection using data mining techniques," *ACM Comput. Surv. (CSUR)*, vol. 50, no. 3, p. 41, Oct. 2017.

[10] X. Du, M. Guizani, Y. Xiao, and H.-H. Chen, "A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1223–1229, Mar. 2009.

[11] Q. Tan, Y. Gao, J. Shi, X. Wang, B. Fang, and Z. Tian, "Toward a comprehensive insight into the eclipse attacks of Tor hidden services," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1584–1593, Apr. 2019.

[12] Z. Tian, Y. Cui, L. An, S. Su, X. Yin, L. Yin, and X. Cui, "A real-time correlation of host-level events in Cyber range service for smart campus," *IEEE Access*, vol. 6, pp. 35355–35364, 2018.

[13] Z. Tian, S. Su, W. Shi, X. Du, M. Guizani, and X. Yu, "A data-driven method for future Internet route decision modeling," *Future Gener. Comput. Syst.*, vol. 95, pp. 212–220, Jun. 2019.

[14] D. Gavrilut, M. Cimpoesu, D. Anton, and L. Ciortuz, "Malware detection using perceptrons and support vector machines," in *Proc. Comput. World, Future Comput., Service Comput., Cogn., Adapt., Content, Patterns*, Nov. 2009, pp. 283–288.

[15] J. Z. Kolter and M. A. Maloof, "Learning to detect malicious executables in the wild," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Aug. 2004, pp. 470–478.

[16] N. Nissim, R. Moskovitch, L. Rokach, and Y. Elovici, "Novel active learning methods for enhanced PC malware detection in windows OS," *Expert Syst. Appl.*, vol. 41, no. 13, pp. 5843–5857, 2014.

[17] R. Islam, R. Tian, L. Batten, and S. Versteeg, "Classification of malware based on string and function feature selection," in *Proc. 2nd Cybercrime Trustworthy Comput. Workshop (CTC)*, Jul. 2010, pp. 9–17.

[18] D. Uppal, R. Sinha, V. Mehra, and V. Jain, "Malware detection and classification based on extraction of API sequences," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2014, pp. 2337–2342.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*, [Online]. Available: https://arxiv.org/abs/1301.3781

[20] R. S. Pirscoveanu, S. S. Hansen, T. M. T. Larsen, M. Stevanovic, J. M. Pedersen, and A. Czech, "Analysis of Malware behavior: Type classification using machine learning," in *Proc. Int. Conf. Cyber Situational Awareness, Data Anal. Assessment (CyberSA)*, Jun. 2015, pp. 1–7.

[21] K. Rieck, P. Trinius, C. Willems, and T. Holz, "Automatic analysis of malware behavior using machine learning," *J. Comput. Secur.*, vol. 19, no. 4, pp. 639–668, 2011.

[22] J. Z. Kolter and M. A. Maloof, "Learning to detect and classify malicious executables in the wild," *J. Mach. Learn. Res.*, vol. 7, pp. 2721–2744, Dec. 2006.

[23] M. G. Schultz, E. Eskin, F. Zadok, and S. J. Stolfo, "Data mining methods for detection of new malicious executables," in *Proc. IEEE Symp. Secur. Privacy S&P*, May 2001, pp. 38–49.

[24] R. Gove, J. Saxe, S. Gold, A. Long, and G. Bergamo, "SEEM: A scalable visualization for comparing multiple large sets of attributes for malware analysis," in *Proc. 11th Workshop Vis. Cyber Secur. (VizSec)*, Nov. 2014, pp. 72–79.

[25] K. Han, B. Kang, and E. G. Im, "Malware analysis using visualized image matrices," *Sci. World J.*, vol. 2014, Apr. Jul. 2014, Art. no. 132713.

[26] A. Long, J. Saxe, and R. Gove, "Detecting malware samples with similar image sets," in *Proc. 11th Workshop Vis. Cyber Secur. (VizSec)*, Nov. 2014, pp. 88–95.

[27] J. Saxe, D. Mentis, and C. Greamo, "Visualization of shared system call sequence relationships in large malware corpora," in *Proc. 9th Int. Symp. Vis. Cyber Secur. (VizSec)*, Oct. 2012, pp. 33–40.

[28] G. E. Dahl, J. W. Stokes, L. Deng, and D. Yu, "Large-scale malware classification using random projections and neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 3422–3426.

[29] J. Saxe and K. Berlin, "Deep neural network based malware detection using two dimensional binary program features," in *Proc. 10th Int. Conf. Malicious Unwanted Softw. (MALWARE)*, Oct. 2015, pp. 11–20.

[30] L. Nataraj, S. Karthikeyan, G. Jacob, and B. Manjunath, "Malware images: Visualization and automatic classification," in *Proc. 8th Int. Symp. Vis. Cyber Secur*, Jul. 2011, p. 4.

[31] L. Nataraj, V. Yegneswaran, P. Porras, and J. Zhang, "A comparative assessment of malware classification using binary texture analysis and dynamic analysis," in *Proc. 4th ACM Workshop Secur. Artif. Intell.*, Oct. 2011, pp. 21–30.

[32] *The Data Set About Malicious Code*. Accessed: Jul. 14, 2019. [Online]. Available: https://github.com/kericwy1337/Datacon2019-Malicious-Code-DataSet-Stage1 and https://github.com/kericwy1337/Datacon2019-Malicious-Code-DataSet-Stage2
[33] L. Van Der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[34] Z. Tian, X. Gao, S. Su, J. Qiu, X. Du, and M. Guizani, "Evaluating reputation management schemes of Internet of vehicles based on evolutionary game theory," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5971–5980, Jun. 2019.
[35] Z. Tian, W. Shi, Y. Wang, C. Zhu, X. Du, S. Su, Y. Sun, and N. Guizani, "Real-time lateral movement detection based on evidence reasoning network for edge computing environment," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4285–4294, Jul. 2019.
[36] Z. Tian, M. Li, M. Qiu, Y. Sun, and S. Su, "Block-DEF: A secure digital evidence framework using blockchain," *Inf. Sci.*, vol. 491, pp. 151–165, Jul. 2019.
[37] S. Li, X. Wu, D. Zhao, A. Li, and Z. Tian, "An efficient dynamic ID-based remote user authentication scheme using self-certified public keys for multi-server environments," *PLoS ONE*, vol. 13, no. 10, Oct. 2018, Art. no. e0202657.

HANGFENG YANG received the B.Sc. degree from the Donghua University of Technology, Nanchang, in 2018. He is currently pursuing the master's degree in computer technology with Guangzhou University, Guangzhou, China. His interests include machine learning and cyber security situational awareness.

SHUDONG LI received the M.S. degree in applied mathematics from Tongji University, China, in June 2005, and the Ph.D. degree from the Beijing University of Posts and Telecommunications, China, in July 2012.

From 2013 to 2018, he held a postdoctoral position at the National University of Defense Technology, China. He is currently an Associate Professor with the Cyberspace Institute of Advanced Technology, Guangzhou University, China. His current research interests include big data and its security, malware identification, information security and cryptography, and the robustness of complex networks. He received the 2015 First prize of Scientific and Technological Progress in Hunan province, China, and the 2018 First prize of Chans Chinese Information Processing Science and Technology, China.

XIAOBO WU received the B.S. degree in computer science from Yantai Normal College, China, in June 2002, and the M.S. degree in computer science and technology from the Dalian University of Technology, China, in June 2008. Her current research interests include protocols analysis, information security, and cryptography.

HUI LU received the B.E. degree in electronic science and technology and the M.S. degree in optical engineering from Southwest Jiaotong University, Chengdu, China, in 2003 and 2006, respectively, and the Ph.D. degree in electromagnetic field and microwave technology from the Beijing University of Posts and Telecommunications, Beijing, China, in 2010.

He was a Research Associate with the Institute of Microelectronics, Chinese Academy of Sciences, China, from 2010 to 2017. In 2017, he joined with the Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, China. His research interests include cyberspace security and intelligent attack-defense.

WEIHONG HAN received the M.S. and Ph.D. degrees from the National University of Defense Technology, China, in 1997 and 2000, respectively. From 2002 to 2017, she was with the National University of Defense Technology. She is currently a Professor with the Cyberspace Institute of Advanced Technology, Guangzhou University, China. Her current research interests include computer networks and network security. She is a member of the Cyber Security Association of China.