RESEARCH ARTICLE OPEN ACCESS

Ransomware Detection by Machine Learning

Mr.Abhijit Pawar, Miss. Pradnya Kapse, Miss.Rutuja Jagtap, Miss.Pooja Dhaigude, Miss.Rupali Deokate

(Assistant Professor, Information Technology, SVPM's College of Engineering Malegaon (bk), Baramati Email:ahpawar@engg.svpm.org.in)
(UG Students, Information Technology, SVPM's College of Engineering Malegaon(bk), Baramati)

Abstract

Today, ransomware has evolved into a severe issue that must be taken into consideration right away in order to prevent moral and financial extortion. Therefore, there is an urgent need for a new technique that can identify and resist this form of attack. The great majority of early detection methods employed a time-consuming dynamic analysis method. The research presented here provides a novel static analysis-based method for locating ransomware. The fundamental component of the proposed method is the deletion of the disassembly stage in favour of direct feature extraction from the raw byte utilising frequent pattern mining, which considerably accelerates detection. The Gain Ratio feature selection approach showed that 1000 features were the optimal quantity for the detection process.The findings demonstrated that, in terms of accuracy and time required, trees with seed numbers of 100 and 1 produced the greatest outcomes. The experimental evaluation showed that the suggested strategy could detect ransomware with a high level of accuracy of 97.74%.*Keywords:* Ransomware detection; Machine learning; Random forest; Cyber security

1.Introduction

Modern attackers use advanced techniques to develop new lucrative virus variants. One of these threats that has gotten a lot of traction recently is ransomware. Ransomware is a persistent, challenging issue with security that cannot be fixed. The method used by this malware is to restrict access to user files by encrypting them and then requesting a ransom in return for the key to unlock them.According to a 2016 analysis by Symantec Corporation, users are forced to pay ransoms totaling hundreds of millions of dollars per year. With the involvement of over 290 companies from various industrial sectors in Europe and the US, Osterman Research and Inc. conducted a survey in 2016. This article examines how to classify ransomware using machine learning by using random forest and features extracted from the file's raw bytes. Different seed and tree sizes have been experimentally investigated in order to develop the best random forest classifier that can correctly detect ransomware.

The major objective is to create an ensemble-based ransomware detection system with a high recall and low false-positive rate that can identify fresh ransomware assaults. Therefore, having a high ability to identify zero-day assaults can help to ease worries about novel attacks.Malware called ransomware is made to prevent a user or business from accessing files on a computer. Cyberattackers put businesses in a situation where paying the ransom is the quickest and least expensive option to recover access to their files by encrypting these files and requesting a ransom payment for the decryption key.

Even if you pay the ransom, you will surely lose your data because some ransomware attacks may be directed especially at your data. Additional causes of ransomware attacks include data loss, information theft, and public revelation of victims' sensitive information. The primary objective of ransomware perpetrators is to extort money from their victims after encrypting data. Numerous ransomware attacks have cost the people they hit with large quantities of money.

1.1 Aim of project

The main goal is to build an ensemble based ransomware detection system that can detect new ransomware attacks, with a high recall and low false-positive rate. Therefore, having a high detection ability of zero-day

attacks can thus reduce concerns related with novel attacks.Ransomware is a malware designed to deny a user or organization access to files on their computer. By encrypting these files and demanding a ransom payment for the decryption key, cyberattackers place organizations in a position where paying the ransom is the easiest and cheapest way to regain access to their files.

1.2 Motivation

Some ransomware attacks may specifically target your data, so even if you pay the ransom, you will undoubtedly lose it. Data loss, theft of confidential information, and disclosure of victims' private information to the public are some more reasons for ransomware attacks. After encrypting data, the main goal of ransomware offenders is to extort money from its victims. Numerous ransomware assaults have drained their victims of substantial sums of money.

1.3.Project Objectives

Malicious software (malware) known as ransomware threatens to publish or prevent access to data or a computer system, typically by encrypting it, unless the victim pays the attacker a ransom price. Since the ransom demand frequently has a deadline, we intended to create a system that could stop frauds and ransomware attacks.

1.4 Application

In this proposed system we will be using RANDOM FOREST TECHNIQUE in which it detect ransomware attacks on individual or industry or organizations as well as prevent the financial loss and some confidential data.

2.Literature Survey

Reference No: 1
Title: Ransomware Detection with Semi-Supervised Learning
Publication: Oxbridge college, kunning university
Author: Xuesong Zhao
Summary: Ransomware is one of the most dangerous cybersecurity risks that businesses and individuals face today. Therefore, creating efficient ransomware detection techniques is imperative. If there is a significant amount of labelled data for training, machine learning techniques can be highly helpful for ransomware detection.

Reference No: 2
Title: Ransomware Prediction Using Supervised Learning Algorithms
Publication: 2014 Second International Conference
Author: Michael D. Ekstrand, John T. Riedl and Joseph A. Konstan
Summary: Ransomware continues to be a concern to both individuals and organisations, and malware has emerged as the most frequent attack vector. One of the major goals of ransomware is to steal money by denying access to system resources like files or the entire system until the ransom is paid. This distinguishes ransomware from other types of malware that aim to reproduce, erase files, excite data, or heavily use system resources.

Reference No: 3
Title: Enhanced Ransomware Detection Techniques using Machine Learning Algorithms
Publication: 2017 International Conference
Author: Khartoum, Sudan
Summary: The growing threat of ransomware attacks is a problem that governments, businesses, and individuals must continuously deal with. Malware of the ransomware variety that encrypts user files and then demands a hefty ransom payment.

Reference No: 4
Title: API-Based Ransomware Detection Using Machine Learning-Based Threat Detection Models
Publication: Jacques
Author: k. Pramodh, Y. Vijayalata
Summary: The approach used to assess an author's writings in order to determine their personality is the main topic of the essay. A computer programme is used to calculate the score for each of the Big-Five personality traits.

Reference No: 5
Title: Android Ransomware Detection using Machine Learning Techniques: A Comparative analysis on GPU and CPU
Author: Fabio Celli ,Bruno Lepri
Summary: Ransomware attacks are carried out by cyber frauds to steal money from its victims by damaging their machines. Because Android is so widely used, assaults on Android-based cellphones are rapidly rising. In this paper, a framework has been proposed in which (1) novel features of Android ransomware are used, (2) machine learning models are used to classify ransomware and safe apps, and (3) a comparative analysis is done to determine the computational time needed by machine learning models to detect Android ransomware. Both locker and crypto ransomware are effectively detected by our suggested methodology. According to the testing findings, the suggested framework can identify Android ransomware with a 99.59 accuracy rate using Logistic Regression on the GPU and CPU, respectively, in 177 milliseconds and 235 milliseconds.

3.System Architecture

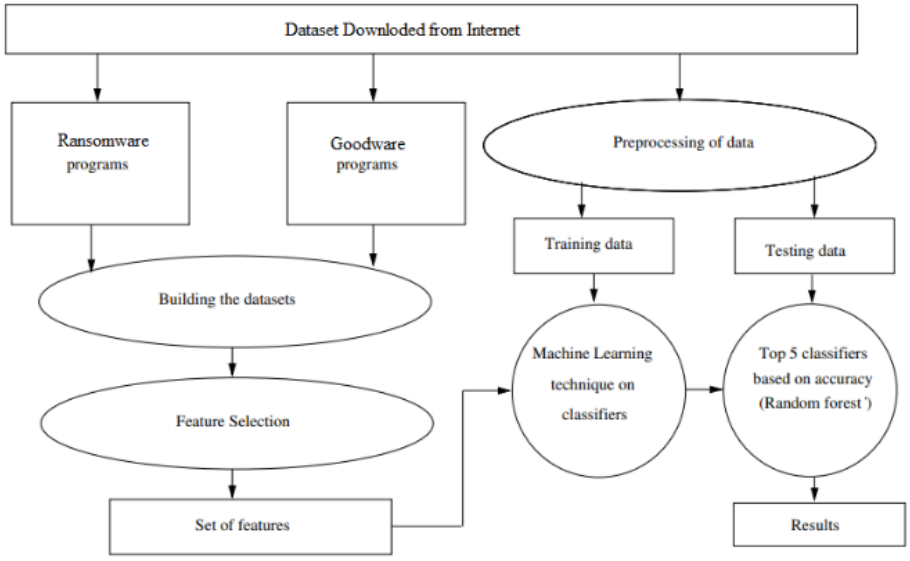

Fig.System Architecture

2.Related works

A ransomware assault exploiting RSA public-key cryptography was launched in September 2013. In 2016, attacks on more than 1,400,000 Kaspersky users across a number of industries led to a global issue (Kaspersky Security Bulletin, 2017). In just one day in 2017, "WannaCry" ransomware infected almost 400,000 machines across 150 nations. (Crowe). As a result, several cyberspace researchers have focused heavily on ransomware detection during the past few years. There are three categories of detection methods, according to general classification: dynamic analysis, static analysis, and hybrid systems that combine the two. Most ransomware detection software utilises behavioural detection, also referred to as "dynamic analysis."It has a high accuracy rate because the ransomware is executed by dynamic analysis. The malicious payload has most likely already been sent at this point because this study takes a while to process and assess. They are also unable to retrieve important API sequences if the malware has left its imprint on the environment. A few researchers have suggested static analysis-based techniques for identifying ransomware attacks. Opcode-based features were employed in a recent study by Zhang et al. to aid in the detection of

ransomware. They used opcode sequences to N-gram sequences and then Term Frequency Inverse Document Frequency as their technique (TF-IDF). To discriminate between ransomware and goodware, five machine-learning techniques, including Decision Tree, Random Forest, K-Nearest Neighbor, Naive Bayes, and Gradient Boosting, were applied. Using random forest, the greatest accuracy of 91.43% was attained. To find ransomware, some researchers utilised a hybrid technique that combines static and dynamic analysis.

The honeypot is another technique that many researchers employ to find ransomware. Moore used a honeypot folder to monitor any changes made to the folder. One researcher developed specific techniques to find ransomware. Kolodenker et al. proposed a PayBreak tool that stores the cryptographic encryption keys in a key vault. These keys are used to decrypt the affected files after a ransomware attack. In a second piece of study, Scaife et al. suggested using the CryptoDrop system, which alerts the user of dubious file activity using a variety of behavioural clues. The honeypot is another technique that many researchers employ to find ransomware. Moore used a honeypot folder to monitor any changes made to the folder. Another method used by many researchers to find ransomware is the honeypot. Moore kept an eye on any modifications in the folder using a honeypot folder. To find ransomware, a researcher created specialised tools. A PayBreak tool that keeps the cryptographic encryption keys in a key vault was proposed by Kolodenker et al. After a ransomware attack, these keys are used to decrypt the impacted files. In a different piece of work, Scaife et al. recommended adopting the CryptoDrop system, which uses a collection of behavioural indications to inform the user during questionable file activity.

3. The proposed method

In order to detect ransomware attacks, this article offers a revolutionary framework that combines static analysis with random forest classifier, one of the most well-known and reliable machine learning techniques. In comparison to other classifiers, this one exhibits substantially more sensible and preferable outcomes when detecting various forms of attacks.Also, this type of classifier has several advantages: - Few input parameters are needed

- The algorithm is resistant to overfitting.
- The variance decreases with increasing in the number of trees without resulting in bias .

The method used in the current study is based on extracting the hierarchical features from the ransomware family because each family of ransomware has a number of characteristics. Because of this, byte-level static analysis has been used, in which the characteristics are directly derived from the executable file's raw bytes (using n-gram features). Additionally, as it deals with bytes, direct features extraction is thought to be quicker and simpler. Three phases make up the preprocessing: normalisation, frequent pattern mining, and feature extraction from raw bytes. For high detection accuracy, the feature extraction method is carried out in a virtual machine (VM) employing 32-bit sliding windows (4-gram) features. The frequent patterns that are connected to important data items are extracted from databases as part of the frequent pattern mining procedure.

The last step is a normalization process, where all frequent patterns are given an equal weight for variance stabilization according to Eq.

$$Nf = \frac{ni,j}{\sum k\ nk,j} \qquad (1)$$

where ∑k nk,j is the total number of features in a file, ni,j is the frequency of particular features, and nf is the normalised frequency.

One of the feature selection techniques for the second stage has been chosen, and it is called Gain Ratio (GR). The role of feature selection in the reduction of feature dimensionality is illustrated by removing the superfluous features and selecting only the most important traits to be included in the current prediction model. The most important stage that follows the feature selection process is the classification stage. The current model utilised the random forest classifier which is one of the supervised learning technique that widely adopted in many studies. This classifier's significant benefit is that it takes less time. Additionally, it has a smaller percentage error while classifying vast amounts of data.

The majority voting for the outcome of the combined forecasts of many decision trees forms the basis for the random forest prediction. The dataset will be divided up into subtrees when the decision tree has been formed based on the optimal combination of variables. Finding the ideal combination of variables is a difficult task, though.

4. Dataset

The dataset consists of 1680 executable files: 840 ransomware executable of different families, and 840 goodware

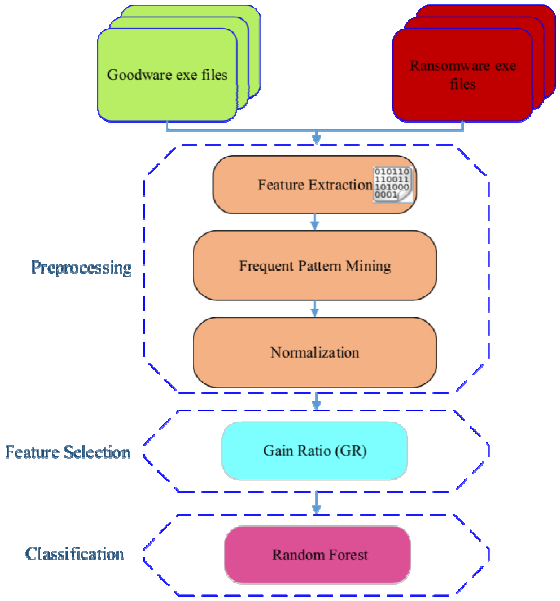

Fig. 1. Shows the flow diagram of the proposed method. It contains the preprocessing, feature selection and the classification technique that used in experiment.

files. The Windows Portable Executable (PE32) ransomware files comprise three different families ; (Cerber (267 samples), TeslaCrypt (315 samples), and Locky (258 samples)) which downloaded from VirusTotal . The goodware files included two types of executable files; first type was collected from windows platform while the other type was collected from Portable Apps platform . Both ransomware and goodware are checked using virustotal.com.Virus Total is a free tool that used to detect whether file is goodware or ransomware file.

The present method was implemented using computer of Core i7 CPU with 8 core, and 16 GB RAM with two systems; Windows 10, and Linux 4.1.

5. Experimental results and discussion

The preprocessed data is split into two groups for training and testing as the initial phase in an experiment. To prevent unbalancing, each group has 50% of the dataset (840 files). The equal number (420 for each) of randomly chosen goodware and ransomware files are included in both groups. Varying tree sizes and different numbers of seeds were used in each tree during the experimental investigation. The optimal design of a random forest classifier that provides high accuracy in identifying ransomware attacks is found using WEKA (a WEKA GUI-based machine learning tool).The common machine learning performance evaluation metrics are used such as False Negative Ratio (FNR), False Positive Ratio (FPR), True Negative Ratio (TNR), True Positive Ratio (TPR), and Accuracy, and F-Measure (the harmonic mean of precision and recall) to evaluate the efficiency of our proposed method, as in following equations:

$$TRP = \frac{TP}{TP + FN}, \quad FRP = \frac{FP}{FP + TN},$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}, \quad Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$F - Measure = 2 * \frac{(Precision * Recall)}{Precision + Recall}$$

where: True Positive (TP): the number of ransomware that is correctly predicted as ransomware.

True Negative (TN): the number of goodware files that are correctly classified as goodware.
False Positive (FP): number of goodware files misclassified as ransomware.
False Negative (FN): number of ransomware which is misclassified as goodware.

5.1. The effect of features dimension

With a seed number of (1) and a tree set of 100, several numbers of features, ranging from 1000 to 7000, have been evaluated to determine the best effective amount of features dimension to create the classifier. Because they provided a very poor detection rate, the remaining size of the features from 100 to 1000 is not included in this result. The classifier accuracy, machine learning performance criteria, and classifier confusion matrix are each displayed in Figs. 2, 3, and Table 1, respectively. The relationship between feature size and accuracy is seen in Fig. 2. The findings clearly show that the 1000 features dimension has the best accuracy, with a score of 97.74%. While this is going on, Figure 2 showed that adding more characteristics has no effect on how accurate the classifier is.The classification model's ROC, Recall, Precision, and F-Measure are shown in Fig. 3. It is clear that the performance for the 1000 features dimension is the greatest in terms of both Precision and Recall. The ROC is around 99.6%, and the F1-measure for 1000 features dimension is above 97.8%. The Confusion Matrix of the current model is shown in Table 1, which suggests that the 1000 features dimension has the best FPR, FNR, TPR, and TNR values, with respective values of 0.043, 0.002, 0.998, and 0.957. In light of these findings, it is clear that a classification model should have a dimension of 1000 characteristics. Therefore, this size of 1000 features will be adopted by all subsequent tests.

5.2. The effect of tree and seed numbers

The current study evaluated various tree sizes ranging from 10-1000 and seed sizes ranging from 1-1000 with regard to the impact of tree and seed numbers. As illustrated in Fig. 4, the testing technique is carried out by limiting the number of seeds to one and altering tree size from 10 to 1000 in accordance with time requirements.It is evident from this figure that classification time is directly proportional with increasing in tree numbers.

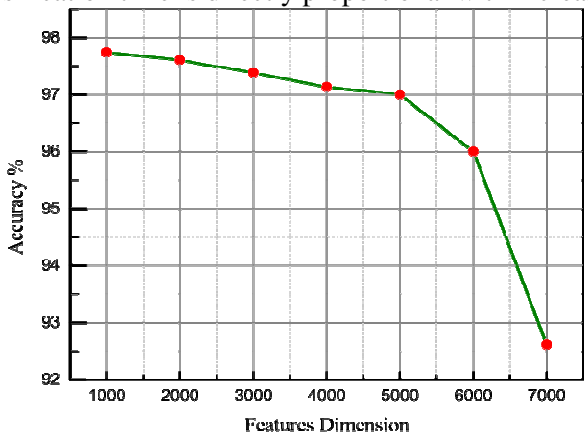

Fig. 2. The accuracy for different features dimension.

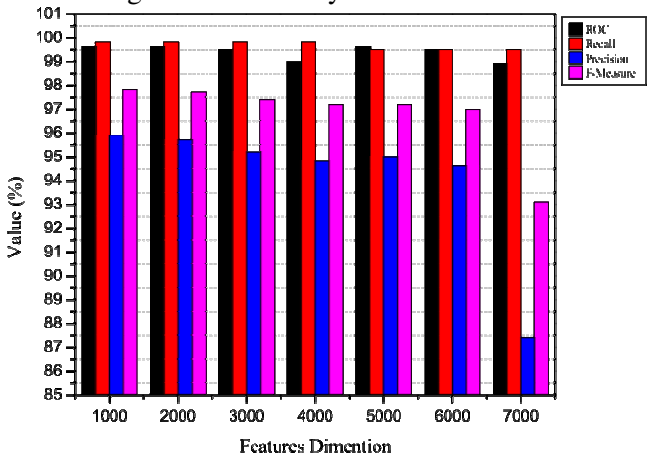

Fig. 3. Recall, F-Measure, Precision, and ROC for different features

Table 1
The TPR, TNR, FPR, and FNR for different features dimension.

Feature dimension	FPR	FNR	TPR	TNR
1000	0.043	0.002	0.998	0.957
2000	0.045	0.002	0.998	0.955
3000	0.05	0.002	0.998	0.95
4000	0.055	0.002	0.998	0.945
5000	0.052	0.005	0.995	0.948
6000	0.057	0.005	0.995	0.943
7000	0.143	0.005	0.995	0.857

The important question raised here is how many trees should be used for classification to get high accuracy within a reasonable amount of time. To address this problem, a number of experiments have been carried out to determine the optimal number of trees for the range of tree sizes from 10 to 100, as shown in Fig. 5, in addition to Confusion Matrix analysis as given in Table 2. With a tolerable time of 1.37 s, it is evident that 100 is the best number among the other tree values in terms of accuracy, recall, F-Measure, precision, ROC, FPR, and TNR.

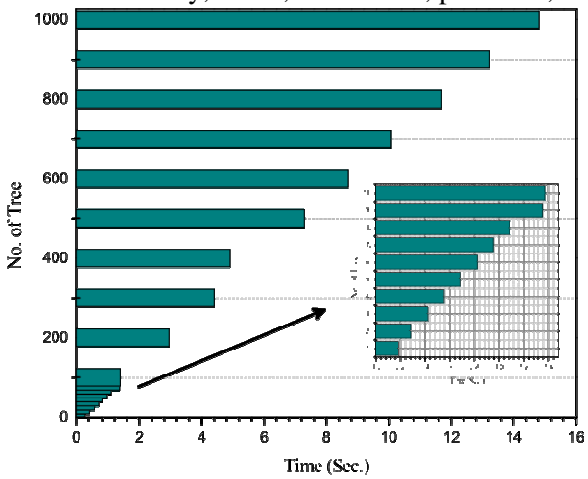

Fig. 4. The time for classification build using 1000 features diminution, the test dataset using different number of tree (10 to 1000) for the random forest classifier.

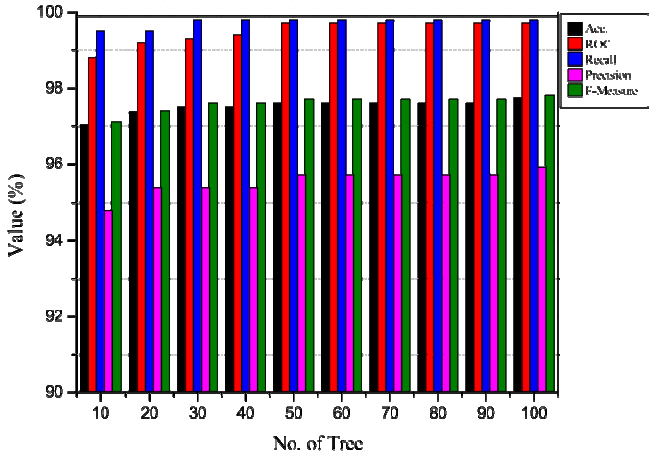

Fig. 5. Accuracy, Recall, F-Measure, Precision, and ROC for different tree number.

Because the accuracy, recall, F-Measure, precision, and ROC are returning the same results as for the 100 tree but taking more time, the remaining number of trees (200 to 1000) are not included in these results. to determine how seed numbers affect the classifier's performance. While the number of seeds varies from 1 to 1000, the number of trees is kept at 100. According to the findings, the best accuracy rate is 97.74% for seed numbers of one, as shown in Fig. 6. This outcome is consistent with the statement that it is usually effective to choose just one seed.

Since the accuracy, recall, F-Measure, precision, and ROC are producing the same results for the remaining trees (200 to 1000), but taking more time, the remaining trees are not included in these results. to determine how the classifier's performance is impacted by seed numbers. The number of trees is kept at 100, while the number of seeds varies from 1 to 1000. The findings are displayed in Fig. 6, which shows that the best accuracy result is 97.74% for seed numbers of one. This outcome is consistent with many who have stated that picking just one seed is usually effective.

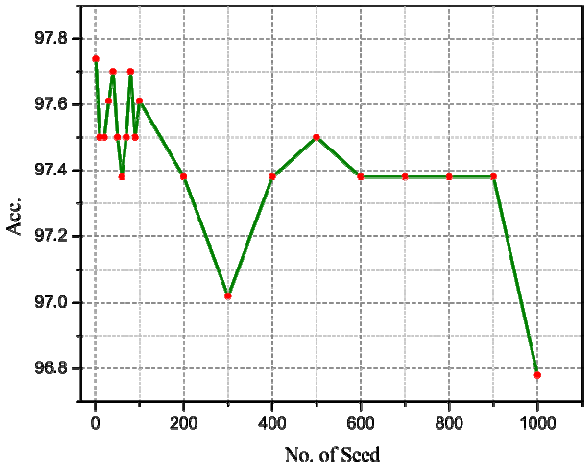

Fig. 6. The accuracy of the random forest classifier using different seed number.

Table 2
The TPR,TNR, FPR, and FNR for different number of tree in the random forest classifier.

No. of tree	FPR	FNR	TPR	TNR
10	0.055	0.005	0.995	0.945
20	0.048	0.005	0.995	0.952
30	0.048	0.002	0.998	0.952
40	0.048	0.002	0.998	0.952
50	0.045	0.002	0.998	0.955
60	0.045	0.002	0.998	0.955
70	0.045	0.002	0.998	0.955
80	0.045	0.002	0.998	0.955
90	0.045	0.002	0.998	0.955
100	0.043	0.002	0.998	0.957

Table 3
The confusion matrix

Classifier type	FPR	FNR	TPR	TNR
Ada Boost M1	0.05	0.14	0.86	0.95
Bagging	0.035	0.062	0.938	0.965
Rotation Forest	0.026	0.031	0.969	0.974
RF	0.043	0.002	0.998	0.957

to establish a foundation for contrasting the estimate metrics gained through testing on an unknown dataset with those produced through k-fold cross-validation. The current study used a 10-fold cross validation method using the whole dataset of 1680 samples. The classifier is iteratively trained using 90% of the training data and checked using

the remaining 10%. After ten iterations, the results are calculated using the average accuracy of all the models. The findings of the 10-fold cross validation and the common categorization measures are shown in Fig. 8 and Table 4, respectively.

The outcomes of Zhang et al., who employed opcode based to express them using n-gram as a feature, are contrasted with those of the current work.This technique need a disassembler to get

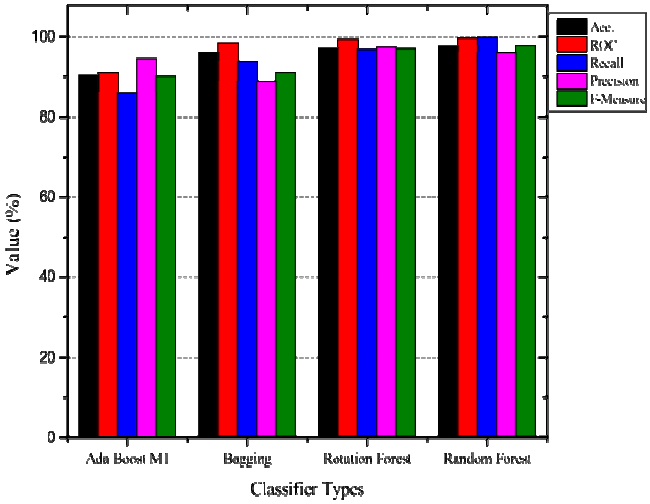

Fig. 7. The standard classification measures of different classifiers.

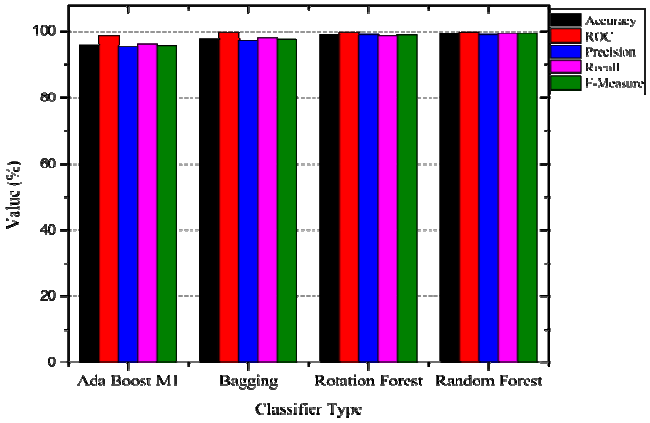

Fig. 8. The standard classification measures of different classifiers when 10fold cross validation has been used.

Table 4

The confusion matrix for 10-fold cross validation of different classifier.

Classifier type	FPR	FNR	TPR	TNR
Ada Boost M1	0.045	0.038	0.962	0.955
Bagging	0.026	0.019	0.981	0.974
Rotation Forest	0.01	0.014	0.986	0.99
RF	0.006	0.007	0.993	0.994

opcode from the file, while the present method has eliminated the disassembly process by extracting the features directly from raw data. The comparison between the suggested approach and the Zhang et al. technique is shown in Table 5. From the comparison, it is clear that the current study exhibits more accuracy than Zhang with a shorter forecast time.

6. Conclusion

This paper offered a method for identifying ransomware attacks based on machine learning (random forest classifier). In the present study, various tree and seed sizes (10–1000 and 1–1000, respectively) were examined. One can draw the following conclusions:

Table 5

The comparison between the proposed technique and Zhang et al. [13] technique.

Method	Features	Dataset (ransomware/ goodware)	Accuracy %	Prediction time in s
Proposed method	Byte level	840/840	97.74	1.37
Zhang et al.	Opcode-based	1787/100	91.43	7.27

1- The experiments show a magnificent performance of random forest classifier with the byte level static analysis for ransomware attack detection.

2- The current analysis emphasized that tree size of 100 with seed size of 1 achieved a high accuracy of (97.74%), high ROC about 99.6%, low FPR (around 0.04), and low FNR (around 0.002) in just 1.37 s time of detection.

3- The features from 100 to 1000 showed bad detection rate. Also, the increasing features number more than 1000 led to a degradation in accuracy.

4- The number of tree from (200 to 1000) showed the same performance of that of 100.

Declaration of competing interest

The authors affirm that they have no known financial or interpersonal conflicts that would have seemed to have an impact on the research presented in this study.

References

[1] H.J. Chittooparambil, et al., A review of ransomware families and detection methods, in: International Conference of Reliable Information and Communication Technology, 2018, pp. 588–597.

[2] I. Osterman Research, Understanding the depth of the global ransomware problem, 2016, http://www.malwarebytes.com/pdf/white-pap ers/UnderstandingTheDepthOfRansomwareIntheUS.pdf.

[3] P. Burnap, et al., Malware classification using self organising feature maps and machine activitydata, Comput. Secur. 73 (2018) 399–410.

[4] X. Luo, Q. Liao, Awareness education as the key to ransomware prevention, Inf. Syst. Secur. 16 (2007) 195–202.

[5] M. Weckstén, et al., A novel method for recovery from crypto ransomware infections, in: 2016 2nd IEEE International Conference on Computer and Communications (ICCC), 2016, pp. 1354–1358.

[6] A. Kharaz, et al.,{UNVEIL}:A large-scale, automated approach to detecting ransomware, in: 25th {USENIX} Security Symposium({USENIX} Security 16), 2016, pp. 757–772.

[7] Y. Takeuchi, et al., Detecting ransomware using support vector machines, in: Proceedings of the 47th International Conference on Parallel Processing Companion, 2018, p. 1.

[8] R. Vinayakumar, et al., Evaluating shallow and deep networks for ransomware detection and classification, in: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp.259–265.

[9] S. Homayoun, et al., Know abnormal, find evil: frequent pattern mining for ransomware threat hunting and intelligence, IEEE Trans. Emerg. Top. Comput. (2017).

[10] A. Tseng, et al., Deep learning forransomware detection, IEICE Tech. Rep. 116 (2016) 87–92.

[11] L. Chen, et al., Towards resilient machine learning for ransomware detection, 2018, arXiv preprint arXiv:1812.09400.

[12] M. Rhode, et al., Early-stage malware prediction using recurrent neural networks, Comput. Secur. 77 (2018) 578–594.

[13] H. Zhang, et al., Classification of ransomware families with machine learning based on N-gram of opcodes, Future Gener. Comput. Syst.90 (2019) 211–221.

[14] J. Baldwin, A. Dehghantanha, Leveraging support vector machine for opcode density based detection of crypto-ransomware, Cyber Threat Intell. (2018) 107–136.

[15] K.P. Subedi, et al., Forensic analysis of ransomware families using static and dynamic analysis, in: 2018 IEEE Security and Privacy Workshops (SPW), 2018, pp. 180–185.

[16] S.K. Shaukat, V.J. Ribeiro, RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning, in: 2018 10th International Conference on Communication Systems&Networks (COMSNETS), 2018, pp. 356–363.
[17] A. Ferrante, et al., Extinguishing ransomware-a hybrid approach to android ransomware detection, in: International Symposium on Foundations and Practice of Security, 2017, pp. 242–258.
[18] C. Moore, Detecting ransomware with honeypot techniques, in: 2016 Cybersecurity and Cyberforensics Conference (CCC), 2016, pp. 77–81.
[19] E. Kolodenker, et al., PayBreak: defense against cryptographic ransomware, in: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, 2017, pp. 599–611.
[20] N. Scaife, et al., Cryptolock (and drop it): stopping ransomware attacks on user data, in: 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS), 2016, pp. 303–312.
[21] A. Continella, et al., ShieldFS: a self-healing, ransomware-aware filesystem, in: Proceedings of the 32nd Annual Conference on Computer Security Applications, 2016, pp. 336–347.
[22] I. Ahmad, et al., Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection, IEEE Access 6 (2018) 33789–33795.
[23] Y. Meidan, et al., Detection of unauthorized IoT devices using machine learning techniques, 2017, arXiv preprint arXiv:1709.04647.
[24] I. Santos, et al., N-grams-based file signatures for malware detection, ICEIS (2) 9 (2009) 317–320.
[25] M.G. Schultz, et al., Data mining methods for detection of new malicious executables, in: Proceedings 2001 IEEE Symposium on Security and Privacy. S & P 2001, 2000, pp. 38–49.
[26] B.M. Khammas, et al., First line defense against spreading new malware in the network, in: 2018 10th Computer Science and Electronic Engineering (CEEC), 2018, pp. 113–118.
[27] B.M. Khammas, et al.,Metamorphic malware detection based on support vector machine classification of malware sub-signatures,TELKOMNIKA (Telecommun. Comput. Electron. Control) 14 (2016).
[28] B.M. Khammas, et al., Feature selection and machine learning classification for malware detection, J. Teknol. 77 (2015).
[29] B.M. Khammas, et al., Pre-filters in-transit malware packets detection in the network, Telkomnika 17 (2019).
[30] I. Ismail, et al., Incorporating known malware signatures to classify new malware variants in network traffic, Int. J. Netw. Manage. 25(2015) 471–489.
[31] I. Santos, et al., Opcode sequences as representation of executables for data-mining-based unknown malware detection, Inform. Sci. 231(2013) 64–82.
[32] M. Shankarpani, et al., Computational intelligent techniques and similarity measures for malware classification, in: Computational Intelligence for Privacy and Security, ed: Springer, 2012, pp. 215–236.
[33] K. Singh,et al., Big data analytics framework for peer-to-peer botnet detection using random forests, Inform. Sci. 278 (2014) 488–497.
[34] K. Singh, B. Nagpal, Random forest algorithm in intrusion detection system: a survey, 2018.
[35] D. Bhalla, Random forest tutorial, 2014, Available: http://www.listen data.com/2014/11/random-forest-with-r.html.
[36] H. Takase, et al., A prototype implementation and evaluation of the malware detection mechanism for IoT devices using the processor information, Int. J. Inf. Secur. 19 (2020) 71–81.
[37] Virus Total - Intelligence Search Engine, http://www.virustotal.com.
[38] https://portableapps.com/apps.
[39] H. Hashemi, et al., Graph embedding as a new approach for unknown malware detection, J. Comput. Virol. Hacking Tech. 13 (2017)153–166.
[40] M.A.M. Hasan, et al., Feature selection for intrusion detection using random forest, J. Inf. Secur. 7 (2016) 129.

