Original Research Paper

Engineering

ANDROID MALWARE DETECTION USING IMPROVISED RANDOM FOREST ALGORITHM

Neelam Research Scholar, Department of Computer Science & Engineering, Punjabi University, Patiala

Charanjiv Singh Saroa* Assistant Professor, Department of Computer Science & Engineering, Punjabi University, Patiala *Corresponding Author

Dr. Gaurav Gupta Assistant Professor, Department of Computer Science & Engineering, Punjabi University, Patiala

ABSTRACT Malignant software or malware keeps on representing a genuine security worry during this computerized age as PC clients, organizations, and governments witness an exponential development in malware assaults. Current malware identification solutions embrace Static and Dynamic investigation of malware marks and behaviour conduct standards that are tedious and ineffectual in distinguishing obscure malwares. Recent malwares use polymorphic, metamorphic and other evasive techniques to vary the malware behaviours quickly and to get sizable amount of malwares. Since new malwares are prevalently variations of existing malwares, AI calculations (MLAs) are being employed to direct a proficient malware examination. This requires extensive feature engineering, feature learning and have representation.. In this paper the actual work was done using individuals five SVM algorithms, decision tree, naïve bays, knn, Random forest to analyze Android detection of malware and suggested that we analyze android malware detection by using majority voting technologies using both SVM and improvisation algorithms..Experiment results shows a proposed approach shows better results as compared to other results.

KEYWORDS : Reverse Engineering, Android, Android Malware Detection, SVM, Random Forest, Machine Learning

I. INTRODUCTION

Malicious software or malware breaches the secrecy and integrity of data and causes unauthorized leakage of information. In recent years, cyber attacks are increasing alarmingly because of the emerging applications of computer and Internet. Hundreds of thousands of new malignant projects are discharged by digital crooks through the Internet trying to take or wreck significant information. Hence, efficient detection and prevention of malicious insiders for protecting valuable data is of critical importance in the computer user community.[1]

Malware detection has faced several drawbacks these past years. Malware analysis mainly relies on two methods for analysis; static and dynamic analysis[3].

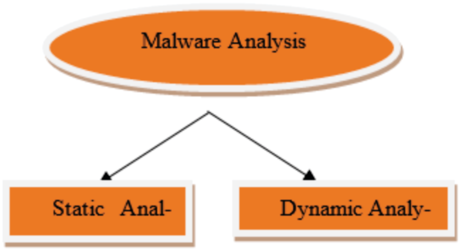

Fig. 1. Types Of Malware Analysis

Static analysis is where a malware analyst disassembles the malware into opcode instructions. Then s/he analyses the sequences of these instructions to draw an outline for the behaviour of the code to determine the files dropped, network connections initiated, processes spawned, etc. This detection technique is time consuming and has been more difficult after malware authors adopted obfuscation techniques. These obfuscation techniques are used to conceal the true code of the malware from the malware analysts; making this method unreliable and inefficient. Examples of obfuscation techniques are polymorphism, where a portion of the code is encrypted with a certain key and decrypted on run-time. A portion of the code is left unchanged and the rest is altered each time. On the other hand, the metamorphic method changes the whole code structure so that no code is the same between variants .

The other approach is dynamic analysis that requires the running of the malware in an automated virtual or emulated environment to detect the maliciousness of the software from the behaviour of the file. This approach is also inefficient due to the need for great computational requirements. Machine learning algorithms can also be used to detect malware.

There are two machine learning methods, supervised and unsupervised machine learning algorithms. Supervised machine learning is used when labeled data is used to train the machine learning model and then used to detect the unlabeled data that is provided. Unsupervised machine learning is used to describe hidden structure or patterns in unlabeled data. In our model, we use the supervised machine learning algorithm; SVM classification. We use opcode sequence trigram along with PE file headers as features. SVM is a discriminative classifier model, which learns a hyperplane from the training dataset for best classification between malicious and benign samples. SVMs unlike other supervised machine learning classification models addresses the drawbacks of overfitting and capacity control and tends to perform better in a variety of scenarios.

II. LITERATURE REVIEW

Mozzamel et al proposed an proficient plan for malware recognition for shielding touchy information from malignant dangers utilizing information mining and AI procedures. Test results shows that the proposed approach gives better execution contrasted with other comparative strategies[1]. Venkatram et al proposed a classical machine learning algorithms (MLAs) and deep learning architectures based on Static analysis, Dynamic analysis and image processing techniques for malware detection and designed a highly scalable framework called ScaleMalNet to detect, classify and categorize zeroday malwares. This framework applies deep learning on the collected malwares from end user hosts and follows a two stage process for malware analysis[2]. Elkawas et al presented our novel methodology in utilizing trigrams and PE record qualities as highlights for malware identification. We adopted a content mining strategy to make our discovery technique progressively powerful to polymorphism and metamorphism. The instruction sequence for critical code in malware on the assembly level is basically the same across malware families. We utilized opcode trigram arrangements as the fundamental component for our AI calculation. We utilized Support Vector Machine (SVM) as our characterizing calculation which is a discriminative classifier model that gives a clear choice whether the anticipated result has a place with the educated class or not[3]. J.Lee et al proposed another approach for this model in which the executables ran in a monitored virtualized environment and the instruction sequences were recorded into basic blocks. These blocks were then used as features depending on the frequency of their appearances in the executable. These features were then classified using the SVM classifier[4]. C.wang et al proposed a client server model that detects malware using opcode instruction sequences with the SVM classifier. The machine learning algorithm resides on the client side and the feature extraction is conducted on the server side[5].Kim et al proposed the primary investigation of the multimodal profound figuring out how to be utilized in the Android malware identification. With our location model, it was conceivable to boost the advantages of incorporating various element types. To assess the exhibition, we completed different examinations with a sum of 41,260 examples. We contrasted the precision of our model and that of other profound neural system models[6]. Park et al proposed the consequences of our experiments to assess the performance of recognizing various sorts of attacks (e.g., IDS, Malware, and Shellcode). We consider the acknowledgment execution by applying the Random Forest calculation to the different datasets that are built from the Kyoto 2006+ dataset, which is the most recent network packet data gathered for creating Intrusion Detection Systems[7]. Jin et al presented SIGPID, a malware identification framework dependent on permission-based examination to adapt to the accelerated increment in the quantity of Android malware. Rather than plucking and evaluating all Android authorizations, we create 3-levels of pruning by mining the consent information to recognize the most noteworthy authorizations that can be viable in recognizing generous and malevolent applications. SIGPID then uses ML-based grouping techniques to order various groups of malware and benign applications[8].

III. PROPOSED METHODOLGY

RF should determine the value during the planning for a minimum extra time in comparison and SVM. The planning is faster and the criteria are less. Exceptions are impossible and therefore it can accommodate the missing attributes and works better with massive repositories and multiple highlights. Therefore, RF is important with a low amount of perceptions for high dimensions. The actual hyper parameters in RF can not be tuned (perhaps besides the amount of trees, trees should be held as much as possible regularly. However, there are still many handles to convert into SVM beyond what might be expected; the selection of the correct parts potential can be uncertain.. RF show improvement over SVM as far as expectation accuracy.

The Stepwise elaborate proposed work:

1.Dataset Collection: The data collected from the online sites.

2. Data Pre-processing: The collected raw information is then pre-processed and converted to a well-defined arff specification, i.e. a csv format. If some missing values are there, it will handle all the missing values by either replacing those values or by removing

3. Classification: Classification is a data mining technique which assigns objects to target classes or categories. The classification aim is to obtain the forecast in the data of the target class for each event. The algorithm tries to establish relations between the attributes / variables to ensure that the result is predictable. SVM and RF algorithms can be used in classifying film reviews received from the Internet in the classification section. Algorithms for detecting and comparing Android malware for the thorough evaluation of performance are used in this work. SVM and RF are graded. Vector support systems are supervised learning model which are used for the separation of classes by calculating the overall margin from both classes with a hyperplanes or set of hyperplanes.

The Random Forest algorithm, on the other hand, acts as a large collection of uncorrelated decision trees. Random Forest is used with a limited number of observations for high-dimensional data.

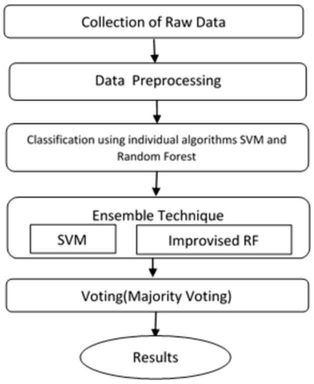

Fig. 2. Flow chart of the proposed work

In our proposed work the modified Random Forest is bagged with the Support Vector Machine using the ensemble learning technique to build a better output architecture compared with single performance algorithms. When used in conjunction, the classification algorithms perform better than if used as single algorithm because each algorithm has its own drawbacks and disadvantages are minimized when the various algorithms are used in combination. The efficiency of the whole learner process improves by using different algorithms, but at the same time a lot of time is used.

Improvised Random Forest

1) Draw a bootstrap sample from the minority class for each iteration in the random forest. Randomly draw from the majority class the same number of cases, replacing them..
2) Induce a classification tree without cutting out from the data to maximum size. Induced by J48 algorithm, the tree will only be searched by a random set of variables at each node, instead of checking all variables to achieve the optimal split in a single node.
3) For the number of times needed, repeat the above two steps. Summarize the ensemble's predictions and estimate the final.

IV. EXPERIMENTAL SETUP

Java is a general purpose,concurrent,class based,object oriented programming language that is specifically designed to have as few implementation dependencies as possible.Java applications are typically combined to byte code that can run on any JVM regardless of computer Architecture.The language devices much as its syntax from C and C++ ,but it has fewer lower facilities than either of them.

V. RESULTS AND DISCUSSION

To study and survey the Malware detection for Android Mobile System by RS Algorithm, following are the parameters used to analyse the performance of proposed prediction model based on accuracy, precision, recall, F-measure and Root mean Square error following are the various parameters used in proposed work:

Accuracy :

Accuracy is one of performance evaluation parameters in which the number of true results such as true positive and true negative among the total number of cases are examined such as true positive, true negative, false positive and false negative.

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

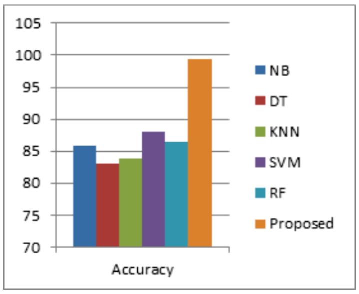

Fig. 3. Accuracy Comparison Chart

Precision:

Precision is defined as the division of retrieved documents that are relevant to the Query.

Precision is defined as:

$$\text{Precision} = \frac{\text{relevant documentss} \cap \text{retrived documents}}{\text{retrived documents}}$$

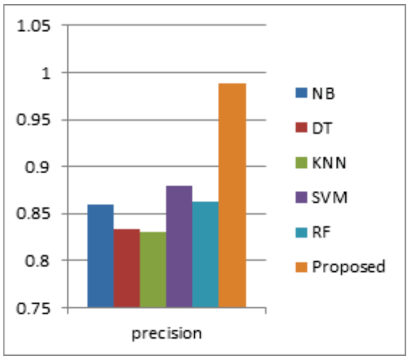

Fig. 4. Precision Comparison Chart

Root Mean Square Error:

The square root of the arithmetic mean of the squares of a set of values.RMSE is also known as RMSD(root mean square deviation). The Root Mean Square Error(RMSE) is the frequently used measure of the difference between values predicted by the model (y) and the values actually observed from the environment(yi) . It can be calculated as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y - yi)^2}{n}}$$

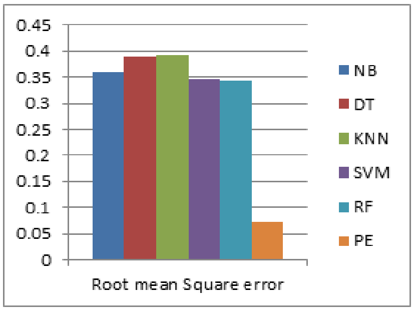

Fig. 5. Root Mean Square Comparison Chart

Recall:

Recall is defined as the fraction of the documents that are relevant to the query that are Successfully retrieved.

$$\text{Recall} = \frac{\text{relevant } documentss \cap \text{retrived documents}}{\text{relevant documents}}$$

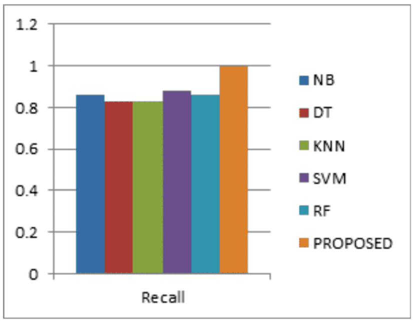

Fig. 6. Recall Comparison Chart

VI. CONCLUSION AND FUTURE SCOPE

The proposed approach will permit Android malware to access sensitive information of mobile devices and detect suspicious activities of android malware. The results of experiments with SVM classifier and improvisement random results show 99% accuracy and improvision are improvisable, random forest and SVM algorithms are used for analyzing android detection malware. In the future, we use the same majority voting strategy, Improvisement Random forests and Help Vector Machine algorithms for some other dataset.

REFERENCES

1. Chowdhury, M., Rahman, A., & Islam, R. (2017). Protecting data from malware threats us.ing machine learning, technique,IEEE ,2017.
2. R, V., Alazab, M., KP, S., Poornachandran, P., & Venkatraman, S., "Robust Intelligent Malware Detection Using Deep Learning", IEEE,2019.
3. Elkhawas, A. I., & Abdelbaki, N. , "Malware Detection using Opcode Trigram Sequence with SVM" , International Conference on Software, Telecommunications and Computer Networks (SoftCOM),2018.
4. J. Dai, R. Guha and J. Lee, "Efficient Virus Detection Using Dynamic Instruction Sequenc-es,", 2009.
5. C Wang, Z. Qin, J. Zhang and H. Yin, "A malware variants detection methodology with an opcode based feature method and a fast density based clustering algorithm", IEEE,2016.
6. Kim, T. G., Kang, B. J., Rho, M., Sezer, S., & Im, E. G. , " A Multimodal Deep Learning Method for Android Malware Detection using Various Features", IEEE,2018
7. Park, K., Song, Y., & Cheong, Y.-G. ,"Classification of Attack Types for Intrusion Detec-tion Systems Using a Machine Learning Algorithm" , IEEE,2018.
8. Li, J., Sun, L., Yan, Q., Li, Z., Srisa-an, W., & Ye, H. (,Significant Permission Identification for Machine-Learning-Based Android Malware Detection. IEEE,2018.
9. Liang, Shuang, & Xiaojiang Du, (2014), "Permission-combination-based scheme for an-droid mobile malware detection", IEEE ,2014.
10. Michal Kedziora, Paulina Gawin, Michal and Ireneusz Jozwiak,"Android Malware De-tection Machine Learning And ReverseEngineering",2018.