International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

AUTOMATED MALWARE DEFENSE: ENSEMBLE LEARNING FOR ANDROID CYBERSECURITY

[1]Prasanna Kumari.Posina, [2] Mr. Yerrabathana Guravaiah
[1]M-Tech, Dept. of CSE Gokula Krishna College of Engineering, Sullurpet
[2]Associate Professor, M-Tech., (Ph.D), CSE Gokula Krishna College of Engineering, Sullurpet

Abstract
Because of its open-source nature and the support it receives from Google, the Android platform currently holds the biggest market share worldwide. Since it is the most widely used operating system in the world, it has attracted the attention of cybercriminals, notably through the widespread distribution of malicious programs. The purpose of this research is to offer an effective machine-learning and deep-learning-based solution for Android Malware Detection. The approach makes use of an evolutionary chi-square algorithm for discriminatory feature selection. For the purpose of training machine learning and deep learning classifiers, selected features from the chi-square algorithm are utilized. The competence of these classifiers to identify malware depending on feature selection is then compared. It has been demonstrated through the results of the experiments that the chi-square method provides the best-optimized feature subset, which assists in the reduction of the feature dimension to less than half of the initial feature set. Machine learning and deep learning-based classifiers, such as RF, ETC, ANN, and CNN, are able to maintain a classification accuracy that is higher than the previous percentage after feature selection. This is accomplished while working on a significantly reduced feature dimension, which positively impacts the computational complexity of learning classifiers. When compared to machine-learning models, deep-learning models have demonstrated the highest level of accuracy in detecting malware. This conclusion is based on the evaluations of our experimental models.

Keywords :- RF, ETC, ANN, CNN, Malware

1. INTRODUCTION

As a result of the fact that cybersecurity is rapidly becoming a primary area of immediate concern for network engineers and computer scientists, it is necessary to find solutions that are satisfactory to several issues [1]. As a consequence of this, the rapid advancements in technology and the intrinsic integrations of these advancements into every facet of lifestyles, as well as the numerous malware applications and targets, become effectively discovered and investigated. Malware of the Android variety is the type of malware that has garnered a great amount of interest in the online community. Android is a popular operating system that now has the majority of the market share for operating systems. As a result of the fact that very few malware apps have more than fifty features, which makes detection a challenging task, malware-invasive tactics are emerging as a means of evading identification. Therefore, it is of the utmost need to develop methods that can effectively cope with the ever-increasing prevalence of malicious software for Android devices, finding it, deactivating it, or removing it. All of these challenges pique the interest of researchers in the field and encourage them to carry out additional research in order to locate malware and effectively control it [2]. There are three different mechanisms that researchers have created in order to identify malware on Android devices. These

mechanisms include dynamic, static, and hybrid analytic methodologies.

Static analysis is a technique that allows for the extraction of information that aids in the identification of damaging performance for applications without the need for an actual application deployment. In contrast, this type of analysis was plagued by code obfuscation techniques, which enable malicious software developers to avoid using static methods. To determine whether or not an application contains malware while it is running, dynamic analysis can be utilized [3]. It is common practice for the static analysis feature to provide the capability of locating the malware element by utilizing the source code, whereas the dynamic analysis feature provides the capability of locating the malware in a runtime environment. Users and developers of Android can be put in a position where they are exposed to risks and dangers that are not essential. The methods of malware detection are discussed in this paper. In order to recognize malicious Android Application Packages (APKs), the machine learning and deep learning model can be utilized for malware detection. This model also incorporates Android Application Packages (APKs) to derive an appropriate set of features.

2. RELATED WORKS

The authors Shaukat et al. [4] come up with a novel method for detecting malware that is related to DL. It produced results that were superior to those of classical combining the advantages of static and dynamic analysis into a single method. The first thing that it does is illustrate a portable executable (PE) file with a variety of colors. The second thing it did was extract deep characteristics from color photos by employing a fine-tuned deep learning technique. In the third place, Malware that belongs to the deep features of SVM is discovered by it.

The authors Geremias et al. [5] revealed a method that can identify malware on Android devices using image-based deep learning. This method is named innovative multi-view Android malware identification. To begin, the apps were evaluated according to the several feature sets that were present in multi-view setups, which resulted in an increase in the amount of data that was offered for categorization. Secondly, the resulting feature set is converted into picture formats while maintaining the fundamental components of the data distribution. This ensures that the data is retained for the purpose of the classifier task. The third point is that created images are portrayed collectively in a single shot, with everything being contained within a preset image channel. This makes it possible to add a DL structure.

Fallah and Bidgoly [6] devised a method for detecting malware that is connected to LSTM. This method has the ability to differentiate between samples of benign and malware, as well as find and detect novel varieties of malware that have not been observed before. An extensive number of tests have been carried out by the author in order to demonstrate the capabilities of the technique that has been provided. These studies include the detection of new malware families, the identification of malware, the identification of malware families, and an evaluation of the minimum amount of time required to locate malware.

Taha and Barukab [7] present a mechanism for the categorization of malware on Android devices that makes use of optimized ensemble learning and is dependent on GA. For the purpose of achieving the highest possible level of

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

accuracy in Android malware classification, the GA was applied for the purpose of optimizing the parameter settings from the RF approach.

Idrees et al. [8] investigate PIndroid, which is a new structure that is built on permissions and intentions and is designed to identify malicious applications for Android. We are aware that PIndroid is the principal solution, which employs a collection of permissions and purposes in addition to Ensemble techniques for the purpose of accurately detecting malware.

3. SYSTEM IMPLEMENTATION

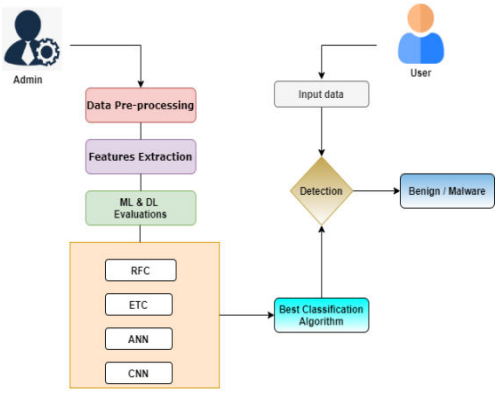

Figure.1 Proposed System Model

Figure.1 depicts the proposed system design and explains briefly in the following:

3.1 Data Pre-processing:

This particular method makes use of the Android malware detection dataset, which is obtained from the Kaggle repository and downloaded before to use. This dataset makes use of 215 features and 3800 occurrences, all of which have the potential to predict a target value such as benign or malicious. In a later stage of the data pre-processing, the dataset will be loaded with the pandas library, and the data frames will be returned. Following that, this system will differentiate between the features that are targeted and those that are independent. Because the target column of this dataset is in string format, the characteristics that are being considered will be changed to numerical type with the use of the label encoding functionality in order to facilitate the understanding of machine learning models.

3.2 Features Selection

During the course of this investigation, the dataset was constructed with 215 features; thus, this system will lower the dimensionality of the dataset by employing the approach of feature selection. A technique for selecting features based on Chi-squared is being utilized here in order to obtain the top-K features. Last but not least, the Android malware detection dataset will be built based on these top-K attributes.

3.3 Models Evaluations

The dataset that has been prepared will be divided, with seventy percent of it being served as a training set and the remaining thirty percent being used as a testing set. The training set will be used to train the RF, ETC, ANN, and CNN portions of the model, and the testing set will be used to evaluate the performance of these components. At long last, each and every model will be returned with their respective performance measures, which include accuracy, precision, recall, and f1-score levels.

3.4 Prediction

Once the outcomes of the experiments have been analyzed, the most accurate model will be chosen for the process of prediction. In this section, the program will supply the input parameters from the prediction form to the most appropriate prediction model in order to forecast the detection findings, such as whether the malware is benign or malicious.

4. METHODOLOGY

RF:

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

The Random Forest (RF) is nothing more than the process of randomly selecting the number of decision trees that are utilized for the purpose of solving classification and regression issues. In this study, the RF is utilized for the purpose of detecting malware on Android devices. As a result, the dataset will be trained using a variety of decision trees, and when the test data points are sent to the RF method, this model will then produce the outcomes that were expected based on the voting systems.

ETC:

Among the several types of ensemble learning algorithms is the additional tree classifier. It creates the collection of decision trees into a set. The choice of the decision rule is made at random during the development of the tree. The Random Forest algorithm is essentially similar to this one, with the exception that the split values are chosen at random. In this case, RF classifiers construct multiple decision trees over bootstrapped subsets of the data, whereas ETCs construct multiple decision trees over the entirety of the dataset.

ANN:

An ANN is a collection of input-output networks that are connected to one another, and each connection has a weight associated with it. One input layer, one or more intermediate layers, and one output layer are the components that make up this structure. Modifying the weight of the link is the method that is utilized in the process of learning neural networks. Iteratively increasing the weight allows for an improvement in the network's performance. During the training process, the weights of the interconnections are optimized until the network reaches the degree of accuracy that was defined. It has several benefits, such as a reduction in the impact of noise and an improved capacity for learning.

CNN:

The convolutional neural network (CNN) is one of the deep learning approaches that is utilized to identify and classify objects in images. In this case, CNN will operate in a different manner, in which neurons will not be connected to every neuron in the layer below them; rather, they will only be connected to neurons that are near them, and all of these neurons will have the same weight. Several layers comprise the CNN, which are what give it its distinctive characteristics. These levels include the convolutional layer, the pooling layer, the rectified linear unit layer (ReLU), and a fully connected layer. In this case, the convolutional layer will be responsible for extracting the feature map, while the ReLU layers will serve as an activation function to reduce the dimensionality of the dataset. Finally, the fully connected layer will be responsible for performing classification on the training dataset. Therefore, the CNN algorithm is utilized in this system to address the categorization issues that are necessary for the Android malware detection system to function properly.

5. EXPERIMENTAL RESULTS

Table 1. Comparative result of the ML and DL approaches.

Algorithms	Accuracy	Precision	Recall	F1 Score
RFC	**92.63**	**91.31**	**92.86**	**91.98**
ETC	**92.89**	**91.56**	**93.24**	**92.28**
ANN	**95.08**	**95.16**	**93.87**	**94.47**
CNN	**95.52**	**95.62**	**94.36**	**94.95**

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
IJARST
ISSN: 2457-0362

Table 1 compares the overall outcomes of the ML and DL model's performance metrics whereas the deep learning models gain slightly enhanced results.

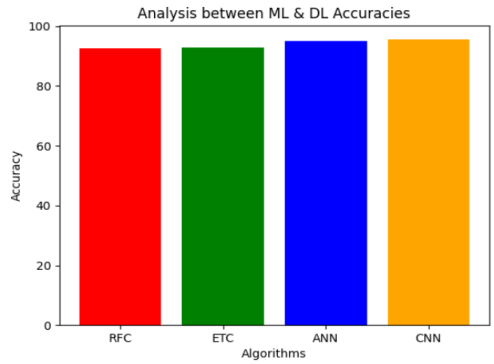

Figure.2 Bar chart with Accuracy of ML and DL algorithms

From figure.2, this research provided the accuracy of ML algorithms RFC is 92.63 percent, ETC is 92.89 percent, the accuracy of DL algorithm ANN is 95.08 percent, and CNN is 95.52 percent. Therefore, the CNN algorithm provided the highest accuracy score.

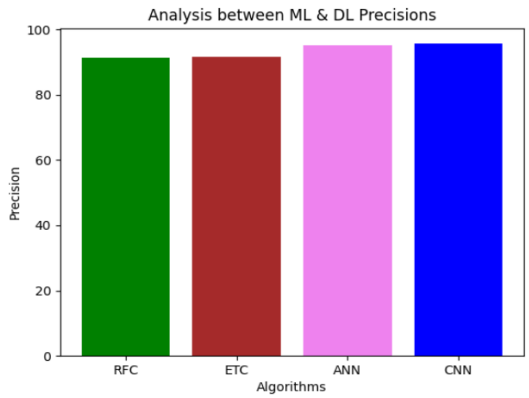

Figure.3 Bar chart with Precision of ML and DL algorithms

From figure.3, this research provided the precision of ML algorithms RFC is 91.31 percent, ETC is 91.56 percent and the precision of DL algorithm ANN is 95.16, and CNN is 95.62 percent. Here, the CNN algorithm provided the highest precision score.

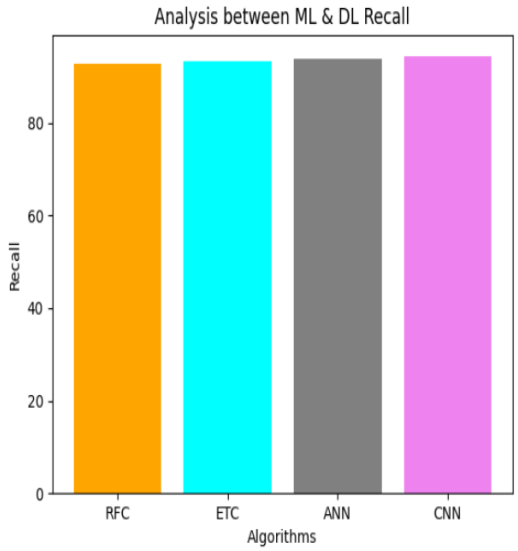

Figure.4 Bar chart with Recall of ML and DL algorithms

From figure.4, this research provided the recall of ML algorithms RFC is 92.86 percent, ETC is 93.24 percent and the recall of DL algorithm ANN is 93.87 percent, CNN is 94.36 percent. So, the CNN algorithm provided the highest recall score value.

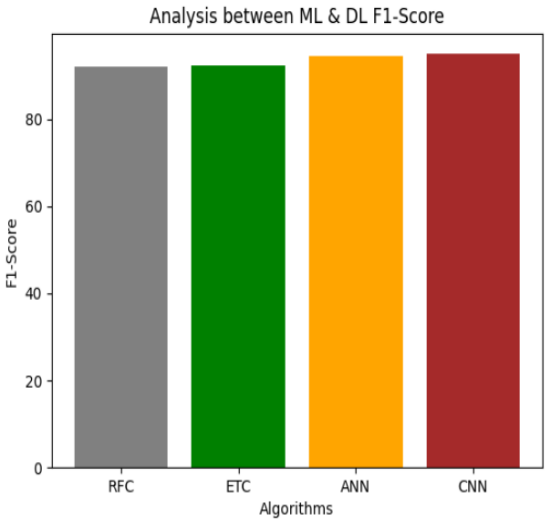

Figure.5 Bar chart with F1-Score of ML and DL algorithms

From figure.5, this research provided the f1-score values of ML algorithms RFC is 91.98 percent, ETC is 92.28 percent and the f1-score value of DL algorithm ANN is 94.47 percent, CNN provided 94.95

percent. So, the CNN algorithm provided the highest f1-score value.

6. CONCLUSION

By constructing an architecture of deep learning approaches, this research has resulted in the introduction of a novel model for detecting malware in programs that run on the Android operating system. Furthermore, a comparison of the various deep learning algorithms will be presented to select the most effective model that is capable of achieving the highest results in identifying malicious software for the Android platform. Noting that the dataset was studied by a static examination of real and realistic samples of malware and benign Android applications, the suggested classifiers RF, ETC, ANN, and CNN were trained on a dataset that was taken from the Android malware dataset. The dataset was used to train the proposed classifiers. According to the findings of the permissions analysis, it would be beneficial to place a greater emphasis on identifying Android malware models. CNN, a deep learning classifier, has beaten all other classifiers, obtaining a high percentage level of accuracy in Android malware detection since it makes use of permissions features.

7. REFERENCES

[1] H. Rathore, A. Nandanwar, S. K. Sahay, and M. Sewak, ‘‘Adversarial superiority in Android malware detection: Lessons from reinforcement learning based evasion attacks and defenses,’’ Forensic Sci. Int., Digit. Invest., vol. 44, Mar. 2023, Art. no. 301511.

[2] L.Hammood, İ.A.Doğru, and K.Kılıç, ‘‘Machine learning-based adaptive genetic algorithm for Android malware detection in auto-driving vehicles,’’ Appl. Sci., vol. 13, no. 9, p. 5403, Apr. 2023.

[3] D.Wang, T.Chen, Z.Zhang, and N. Zhang, ‘‘A survey of Android malware detection based on deep learning,’’ in Proc. Int. Conf. Mach. Learn. Cyber Secur. Cham, Switzerland: Springer, 2023, pp. 28–242.

[4] K. Shaukat, S. Luo, and V. Varadharajan, ‘‘A novel deep learning-based approach for malware detection,’’ Eng. Appl. Artif. Intell., vol. 122, Jun. 2023, Art. no. 106030.

[5] J. Geremias, E. K. Viegas, A. O. Santin, A. Britto, and P. Horchulhack, ‘‘Towards multi-view Android malware detection through image-based deep learning,’’ in Proc. Int. Wireless Commun.MobileComput. (IWCMC), May 2022, pp. 572–577.

[6] S. Fallah and A. J. Bidgoly, ‘‘Android malware detection using network traffic based on sequential deep learning models,’’ Softw., Pract. Exper., vol. 52, no. 9, pp. 1987–2004, Sep. 2022.

[7] A. Taha and O. Barukab, ‘‘Android malware classification using optimized ensemble learning based on genetic algorithms,’’ Sustainability, vol. 14, no. 21, p. 14406, Nov. 2022.

[8] F. Idrees, M. Rajarajan, M. Conti, T. M. Chen, and Y. Rahulamathavan, ‘‘PIndroid: A novel Android malware detection system using ensemble learning methods,’’ Comput. Secur., vol. 68, pp. 36–46, Jul. 2017.