JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 9, ISSUE 04, 2022

Effective Malware Detection Using Machine Learning

P. Anusha,[1]B. Lakshmi Bhavani[2], M.V.N.Lakshmi Revathi [3], U.Sri Vimal Vishnu[4], O.Naga Swetha[5]

[1]Asst. Professor, Department of Computer Science and engineering

[2,3,4,5]Student, Department of Computer Science and engineering

[1,2,3,4,5]QIS College of Engineering and Technology

Abstract-Malicious programmes are a major problem on Android, which enables developers to take full advantage of the mobile OS. Malicious applications are a major concern as the Android ecosystem expands. Machine learning-based algorithms are among the most accurate at identifying malware of the many methods available. On the other side, the growing quantity of apps creates an ideal environment for malicious users to create new malware and distribute it via the Google Android Market or other third-party marketplaces as a legitimate programme. So it is necessary to do malware analysis or reverse engineering on such harmful programmes that threaten Android systems seriously. Using an evolutionary genetic algorithm and a grid search cv method, this research provides an effective machine-learning-based technique for Android malware detection. Before and after feature selection, machine learning classifiers are compared in terms of their ability to correctly identify Malware.

Keywords— Malware, Identification, Hackers, Attackers.

I. INTRODUCTION

Google Playstore, the official Android app store, and third-party app shops allow users to download Android apps for free. Because of Android's open source nature and popularity, malware authors are rapidly producing dangerous Android apps. As a result of Google Playstore's numerous efforts to prevent malicious apps from making their way into the general market, they still find their way into the hands of consumers and cause harm to them by exploiting personal information such as their phone book, email accounts, GPS location information, and more. So it is necessary to do malware analysis or reverse engineering on such harmful programmes that threaten Android systems seriously.

II. RELATEDWORKS

Generic Algorithms may be used to decrease feature dimensions to less than half of the original feature-set so that they can be provided as input to machine learning classifiers for training with reduced complexity while keeping their accuracy in malware classification. For feature selection, Genetic Algorithm is a heuristic search methodology based on fitness function instead of an exhaustive method that tests for 2N distinct combinations of N features. Two machine learning algorithms, the Support Vector Machine and the Neural Network, are trained using the genetic algorithm's optimal feature set.

According to Wu et al(2014), the static malware examination and the element malware research were carried out using an artificial immune-based smartphone malware detection model (P-MDM) that is capable of protecting humans from sickness by critters. Antigens are created by separating the malware's static and dynamic identifiers and encoding them using the technique of genuine esteemed vector encoding. Whenever an identifier encounters opposition from inside, it changes into a developed one. Following the simplification of producing identifiers that makes use of the clonal determination calculation, finder posterity is being formed with more affection. Bat-Erdene et al. (2017) developed a method for visualising the packing methodology of an unknown packed executable using twenty malicious files and the same number of benign files as testing samples. After examining the executable's entropy estimates, the researchers changed the entropy estimation of the memory area they were interested in into conventional representations. Authors used a symbolic aggregate approximation (SAX) that is known for its usefulness in situations when a large amount of data is changed. Using the controlled learning order tactics, such as bolster vector machines and credulous Bayes, the authors then ask for pictures to be made available for use in detecting urgent computations. It was found that the approach was able to distinguish pressing computations of the offered executable with a large quantity of precision and accuracy after analysing 324 pressed kindhearted projects and 326 filled malware programmes.

Cui et al. (2015) conducted a research to determine the best framework for packet identification in a cloud setting. Using information mining techniques, this method identifies harmful mobile malware by studying its bundles. A common problem with traditional procedures is their tendency to produce unintended abnormalities. The government has set up a gadget that may be used to alert customers if their computers are infected with malware. Withdrawal grouping, a support method, was devised to help implement this strategy. This method relies on prior

ISSN- 2394-5125 VOL 9, ISSUE 04, 2022

knowledge to decrease the dataset's size. An additional placement plan for several modules also aided in the framework's prevision. Calculations such as Naive Bayes and Decision Tree are used to determine the effects of implementing this strategy's recommendations.

An innovative calculation by Fan, Ye, and Chen (2016) revealed a vengeful quintal case calculation based on arrangement mining, and afterwards an All-Nearest-Neighbor (ANN) was built for a malicious location in the established samples. Using the suggested sequential case of the mining approach and ANN classifier, the information mining structure was constructed to adequately highlight current malware testing that are malicious. Even though a valid dataset was analysed for exploratory purposes, the recognition structure was evaluated by accumulating data. When comparing their structure to other trading information mining based on discovery methodologies, the prospective test shows that theirs is superior in terms of distinguishing fresh spiteful executables

A new diagram mining technique, developed by Hellal and Ben Romdhane (2016), uses static evaluation to identify malware changes while masking the current faults. Minimal contrast frequent subgraph miner (MCFSM) is another unique computation that researchers presented (MCFSM). For this method's purpose, it had to be used to distinguish between benign programmes and spiteful ones that could be distinguished with accuracy, despite the fact that they were both motivated by malice. One advantage of this method is that it is quite accurate in its detection, with just a small percentage of false positives.

It was established in Martn, Menéndez, and Camacho (2016) that outsiders could easily get around the impact of the disguise tactics since they were not obfuscated. The researchers work together to construct a classifier for certain activities that are defined by what outsiders term "bunches" via bunching and multi-target progress. In the opinion of the analyzer, every non-discriminatory example of noxious or appropriate cleaning activities is recognised by the groups. MOCDroid has a 95.15% accuracy rate in testing and a 1.69 percent false positive rate for apps that have been pulled from the wild, with the ability to evade all the antivirus engines from CirusTotal.

Another technique was devised by Santos et al. (2013) in an effort to identify the hidden families of malware. The approach is built on the idea that opcode groups occur often. The researchers were able to devise a method for keeping track of the importance of each opcode and determining the frequency with which certain opcode groupings occur. It was also confirmed via an approved trial run that this new technique was suitable for identifying cryptic malware.

Disadvantages of existing system:

- No Authentication and Security
- Time complexity
- Less accuracy

III. PROPOSED SYSTEM ARCHITECTURE

Tkinter is used as a front-end framework in the current system. Authentication and security, on the other hand, are unspecified. High security for the application against unauthorised usage is provided by the powerful front end framework Flask and an extension of user and admin. In addition, the grid search cv algorithm will be applied to the current Genetic algorithm in order to improve accuracy. It was applied to xb boost, adb boost, and logistic regression to reduce the errors in the final output results.

Acquiring dataset : Acquiring the dataset is the first step in data preprocessing in machine learning. To build and develop Machine Learning models, you must first acquire the relevant dataset. This dataset will be comprise of data gathered from multiple and disparate sources which are then combined in a proper format to form a dataset. Dataset formats differ according to use cases. For instance, a business dataset will be entirely different from a medical dataset. While a business dataset will contain relevant industry and business data, a medical dataset will include healthcare-related data.

Pre-processing :Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

Training the dataset :

Training set (usually 70-80% of data): Model learns on this.

Validation set (usually 10-15% of data): Model hyper parameters are tuned on this

Test set (usually 10-15% of data): Models' final performance is evaluated on this.

Genetic Algorithm

1. Start the algorithm defining an initial set of population generated randomly.

2. Assign a fitness score calculated by the defined fitness functionfor genetic algorithm.

ISSN- 2394-5125 VOL 9, ISSUE 04, 2022

3. Selection of parents : Chromosomes with god fitnesss scores aregiven preference over others to produce next generation.
4. Perform crossover and mutation operations on the selectedparents with the given probability of crossover and mutation for generation of off-springs.
5. Repeat the steps 2 to 4 iteratively till thre convergence is metand fittest chromosome from population, that is, optimal feature subset is resulted

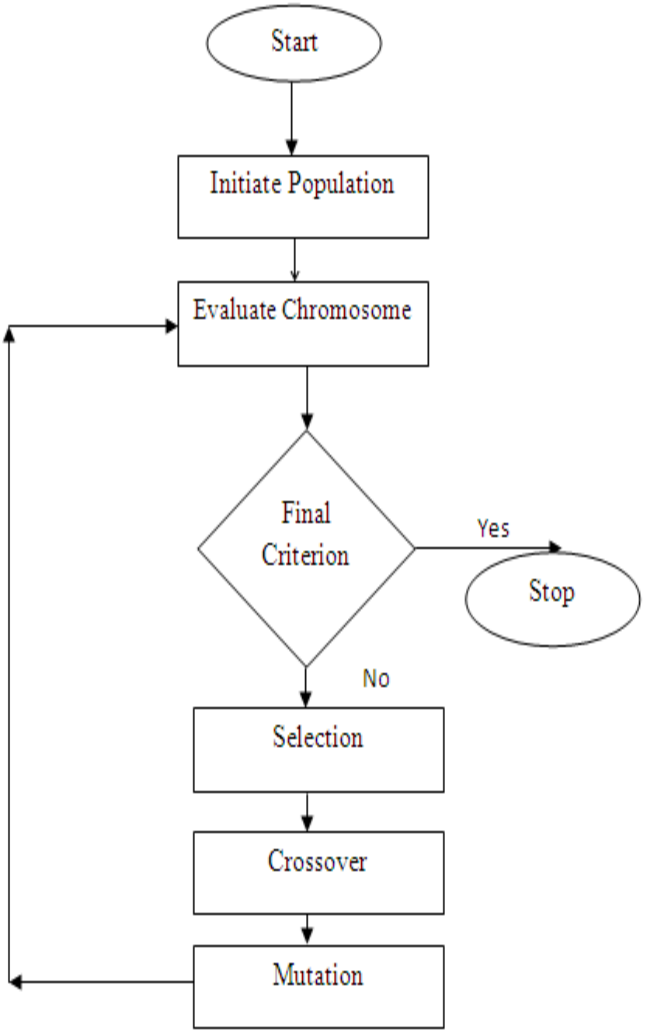

Fig.1 Genetic Algorithm

OpenGrid Search Cv

1 .Take the input and Prepare the database.
2 . Initialize grid search space construction
3 . Identify the model's hyperparameters to optimize, and thenwe select the hyperparameter values that we want to test.
4 . Start the performance evaluation and Asses error score for eachcombination in the hyperparameter grid.
5 . Select the hyperparameter combination with the best error metric.
6 . Try again with these parameters

Advantages of Proposed system are

- Computational Cost is low
- Authentication and Security

ISSN- 2394-5125 VOL 9, ISSUE 04, 2022

- Parallelism
- Requires less information

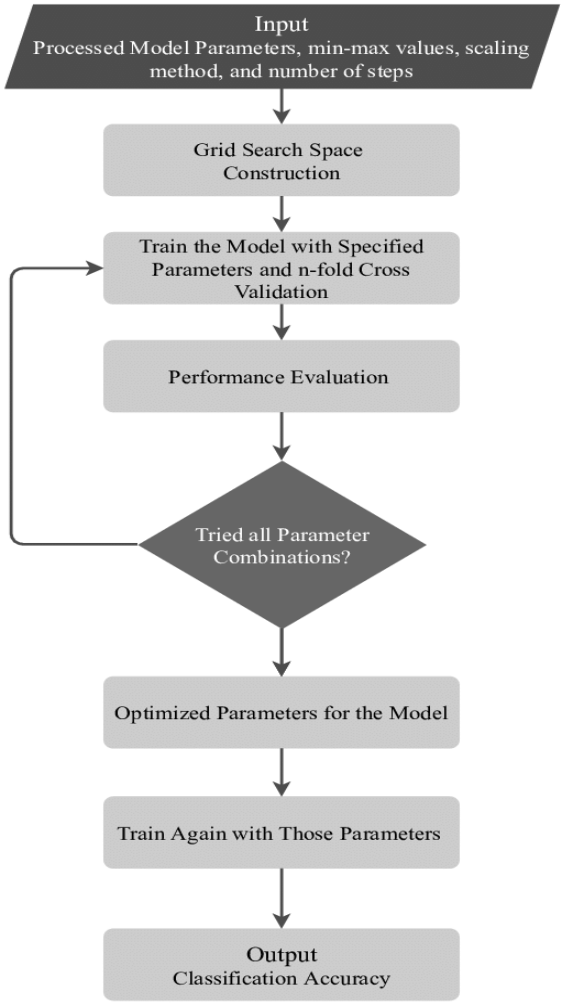

Fig.2**OpenGrid Search Cv**

IV. RESULTS AND DISCUSSION

There are 40,000 APKs in the dataset: 20,000 malicious ones and 20,000 non-malicious ones. Reverse engineering is used to extract features from the APKs. Malware (represented by zero) and Goodware (represented by one) are included in a CSV containing 99 features (represented by 1). Genetic Algorithm has been utilised to find the most optimal feature subset for this study. A Support Vector Machine or a Neural Network classifier may be trained using the characteristics chosen via Genetic Algorithm. Support Vector Machine's parameters are as follows: The kernel function is RBF, and the number of cross-validation folds is 10. The feed-forward neural network has a size 40 hidden layer architecture.An Intel Xeon(R) Silver 4114 CPU with a clock speed of 2.20 GHz and RAM of 64 GB was used to evaluate the algorithms. Before and after feature selection, the performance of these two classifiers in differentiating between Malware and Goodware is compared.For each classifier, the Genetic algorithm selects a subset of characteristics to be used for classification, and these features are shown in Table I. Both classifiers retain a considerable amount of accuracy, as shown in Table I, despite large reductions in the number of features.Performance metrics for Support Vector Machine and Neural Network classifiers are shown in Tables II and III before and after feature selection, respectively. If you look at performance metrics, both Support Vector Machine and Neural Network when combined with Genetic Algorithm for feature selection perform significantly

ISSN- 2394-5125 VOL 9, ISSUE 04, 2022

well without sacrificing much accuracy while working in less vector space (less than half of the original feature-set), thus reducing classifier training time complexity, as can be seen in this graph.

TABLE I. FEATURE SELECTED BY GENETIC ALGORITHM FOR DIFFERENT CLASSIFIERS AND ACCURACY OBTAINED WITH SELECTED FEATURES

Algorithm	No of features before feature selection	AUC	Features Selected	AUC
Support Vector Machine	99	.9891	33	.9803
Neural Network	99	.9876	40	.9828

TABLE II. PERFORMANCE METRICS OF SUPPORT VECTOR MACHINE CLASSIFIER

Performance Metrics	With 99 features	With 33 features (post feature selection)
Sensitivity (%)	95.5	94.6
Specificity (%)	97.6	95.4
Accuracy (%)	96.6	95.0
Training Time Complexity (secs)	22.92	10.20

TABLE III. PERFORMANCE METRICS OF NEURAL NETWORK CLASSIFIER

Performance Metrics	With 99 features	With 40 features (post feature selection)
Sensitivity (%)	95.6	94.3
Specificity (%)	94.9	93.9
Accuracy (%)	95.2	94.1
Training Time Complexity (secs)	8.57	3.76

V. FUTURE SCOPE AND CONCLUSION

As the number of threats to Android platforms grows daily, spreading mostly via malicious apps or malware, it is critical to build a framework that can accurately identify such malware. Machine learning-based techniques are utilised when signature-based approaches fail to identify new varieties of malware that pose zero-day risks. The suggested technique uses an evolving Genetic Algorithm to find the most efficient feature collection for training

ISSN- 2394-5125 VOL 9, ISSUE 04, 2022

machine learning algorithms. Support Vector Machine and Neural Network classifiers retain a respectable classification performance of more than 94% despite operating on a reduced dimension feature-set, hence lowering the complexity of the classifiers' training. When combined with Genetic Algorithm, future study may take use of bigger datasets for better results and an examination of the impact on other machine learning techniques.

REFERENCES

[1] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, and K. Rieck,"Drebin: Effective and Explainable Detection of Android Malware inYour Pocket," in *Proceedings 2014 Network and Distributed SystemSecurity Symposium*, 2014.

[2] N. Milosevic, A. Dehghantanha, and K. K. R. Choo, "Machine learningaided Android malware classification," *Comput. Electr. Eng.*, vol. 61,pp. 266–274, 2017.

[3] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-An, and H. Ye, "SignificantPermission Identification for Machine-Learning-Based AndroidMalware Detection," *IEEE Trans. Ind. Informatics*, vol. 14, no. 7, pp.3216–3225, 2018.

[4] A. Saracino, D. Sgandurra, G. Dini, and F. Martinelli, "MADAM:Effective and Efficient Behavior-based Android Malware Detection andPrevention," *IEEE Trans. Dependable Secur. Comput.*, vol. 15, no. 1,pp. 83–97, 2018.

[5] S. Arshad, M. A. Shah, A. Wahid, A. Mehmood, H. Song, and H. Yu,"SAMADroid: A Novel 3-Level Hybrid Malware Detection Model forAndroid Operating System," *IEEE Access*, vol. 6, pp. 4321–4339, 2018.

[6] T. Kim, B. Kang, M. Rho, S. Sezer, and E. G. Im, "A Multimodal DeepLearning Method for Android Malware Detection using VariousFeatures," vol. 6013, no. c, 2018

[7] A. Martin, F. Fuentes-Hurtado, V. Naranjo, and D. Camacho, "EvolvingDeep Neural Networks architectures for Android malwareclassification," *2017 IEEE Congr. Evol. Comput. CEC 2017 - Proc.*, pp.1659–1666, 2017.

[8] X. Su, D. Zhang, W. Li, and K. Zhao, "A Deep Learning Approach toAndroid Malware Feature Learning and Detection," 2016 IEEE Trust.,pp. 244–251, 2016.

[9] K. Zhao, D. Zhang, X. Su, and W. Li, "Fest : A Feature Extraction andSelection Tool for Android Malware Detection," *2015 IEEE Symp.Comput. Commun.*, pp. 714–720, 4893.

[10] A. Feizollah, N. B. Anuar, R. Salleh, and A. W. A. Wahab, "A reviewon feature selection in mobile malware detection," *Digit. Investig.*, vol.13, pp. 22–37, 2015.

[11] A. Firdaus, N. B. Anuar, A. Karim, M. Faizal, and A. Razak,"Discovering optimal features using static analysis and a genetic searchbased method for Android malware detection *," vol. 19, no. 6, pp. 712–736, 2018.

[12] A. V. Phan, M. Le Nguyen, and L. T. Bui, "Feature weighting and SVMparameters optimization based on genetic algorithms for classificationproblems," *Appl. Intell.*, vol. 46, no. 2, pp. 455–469, 2017.

Authors Profile

P.Anushacurrently working as Assistant professor of Computer Science & Engineering in Qis college of Engineering and Technology (Autonomous& NAAC 'A'Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist. Affiliated to Jawaharlal Nehru Technological University, Kakinada.

U.Sri Vimal Vishnu pursuing B.Tech in the department of Computer Science & Engineering from Qis college of Engineering and Technology (Autonomous & NAAC'A'Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist. Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2018-2022 respectively.

ISSN- 2394-5125 VOL 9, ISSUE 04, 2022

M.V.N.Lakshmi Revathi pursuing B.Tech in the department of Computer Science & Engineering from Qis college of Engineering and Technology (Autonomous & NAAC'A'Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist. Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2018-2022 respectively.

B.Lakshmi Bhavana pursuing B.Tech in the department of Computer Science & Engineering from Qis college of Engineering and Technology (Autonomous & NAAC'A'Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist. Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2018-2022 respectively

O.Naga Swetha pursuing B.Tech in the department of Computer Science & Engineering from Qis college of Engineering and Technology (Autonomous & NAAC'A'Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist. Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2018-2022 respectively.