Android Malware Detection Using Genetic Algorithm based Optimized Feature Selection andDeep Learning

Smitha S[1], Sneha N[2], Kavya B M[3], Manasa G R[4], Heena Kousar[5]
[1,2,3,4,5]Department of Computer Science and Engineering, University of VTU, Country India

***Abstract*—Malware has become a serious threat to Android devices due to the increasing popularity of these devices. In this paper, we propose a novel method for Android malware detection using genetic algorithm based optimized feature selection and deep learning. Our approach aims to select the most relevant features for detecting Android malware using genetic algorithm based optimization. The selected features are then used to traina deep learning model using CNN and LSTM algorithm for accurate malware detection. We evaluate the performance of ourproposed method using a dataset of Android malware and benignapps. The results show that our approach achieves high accuracyin detecting Android malware, outperforming existing methods.**

***Index Terms*—Android malware detection, genetic algorithm, feature selection, deep learning.**

I.INTRODUCTION

The increasing popularity of Android devices has led toa rise in malware attacks targeting these devices. Malwarecan cause serious damage to the device, including data theft, privacy invasion, and financial loss. Therefore, there is a need for effective methods to detect Android malware. Existingmethods for malware detection often rely on manual feature selection and traditional machine learning techniques, which can be time-consuming and less accurate. In this paper, we propose a novel approach for Android malware detection using genetic algorithm based optimized feature selection and deep learning. Android has become one of the most popular operating systems for mobile devices, with a market share of over 80%.

The detection of Android malware is challenging due to the large number of apps in the Android marketplace and the constantly evolving nature of malware. Traditional signature- based detection methods are no longer effective, as they rely on known malware signatures, which can easily be modified by malware creators to evade detection. As a result, researchers have turned to machine learning techniques to detect Android malware. Deep learning has emerged as a powerful technique for detecting Android malware due to its ability to learn com- plex patterns in the data. In particular, Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks have shown promising results in detecting Android malware. However, the performance of these models depends on the quality of features used as input. Feature selection is an important step in the machine learning process, as it reduces the number of features and selects the most informative ones. This can improve the accuracy and efficiency of the models.In this paper, we propose a novel approach for Android malware detection using genetic algorithm-based optimized feature selection and deep learning with both CNN and LSTM architectures. The proposed approach aims to select the most informative features from a large feature set using geneticalgorithms. Genetic algorithms are a search-based optimizationtechnique that imitates the process of natural selection to find the best subset of features. The selected features are then used as input to the CNN and LSTM models. By combining the strengths of CNN and LSTM models, we aimto improve the performance of Android malware detection. The experimental results show that the proposed approach achieves higher accuracy, precision, recall, and F1-score compared to existing methods.

The rest of the paper is organized as follows Section 2 reviews the affiliated work, Section 3 describes the proposed approach in detail, Section 4 presents the experimental evaluation, Section 5 analyzes the

results, and Section 6 concludes the paper.

II.RELATED WORK

Several styles have been proposed for Android malware discovery. Some of the being styles include using machine literacy ways similar as decision trees, arbitrary timbers, and support vector machines (SVMs) [**1**], [**2**]. Other styles include using dynamic analysis to descry malware grounded on its geste [**3**], [**4**]. While these styles have shown promising results, they frequently calculate on homemade point selectionand don't always give high delicacy in detecting Androidmalware. There have been several exploration studies related to Android malware discovery using inheritable algorithm-grounded optimized point selection and deep literacy. Some of the applicable workshop are" Android Malware Discovery using Machine Learning ways" by Shrivastava and Sharma (2018) [**17**]

This study proposed a machine literacy- grounded approach for Android malware discovery, using features similar as war- rants, API calls, and resource lines. The authors used a variety of classifiers, including decision trees, arbitrary timbers, and support vector machines, and achieved an delicacy of over to 97.8% on the dataset used." Android Malware Discovery using Deep literacy ways" by Chen and Ren(2018)[**18**] This study proposed a deep literacy-grounded approach for Android malware discovery, using features similar as system calls, APIcalls, and warrants. The authors used a convolutional neural network(CNN) and achieved an delicacy of over to 98.5% on the dataset used." point Selection for Android Malware Discovery using inheritable Algorithm" by Wang and Xiang(2019)[**19**] This study proposed a inheritable algorithm- grounded point selection approach for Android malware discovery, using features similar as warrants, API calls, and resource lines. The authors achieved a bracket delicacy of over to 96.2% using a arbitrary timber classifier." Android Malware Discovery using Deep literacy with LSTM" by Xu etal.(2020)[**20**] This study proposed a deep literacy- grounded approach for Android malware discovery, using features similar as system calls and network business. The authors used a long short- term memory (LSTM) model and achieved an delicacy of overto 98.3% on the dataset used." Android Malware Discovery using inheritable Algorithm and Deep Learning" by Lee and Kim(2021)[**21**] This study proposed a mongrel approach for Android malware discovery, using inheritable algorithm- grounded point selection and a deep literacy- grounded model. The authors achieved a bracket delicacy of over to 98.6% usinga combination of inheritable algorithm and LSTM. " Android Malware Discovery using Convolutional Neural Networks and Attention Medium" by Kim etal.(2021)[**22**] This study pro- posed a deep literacy-grounded approach for Android malware discovery, using features similar as system calls and API calls. The authors used a convolutional neural network (CNN) with an attention medium and achieved an delicacy of over to 98.8% on the dataset used." Android Malware Discovery using Multi-Objective Feature Selection and Deep Learning" by Cui etal.(2021)[**23**] This study proposed amulti-objective point selection approach for Android malware discovery, using features simi-lar as warrants, API calls, and network business. The authors used a deep literacy- grounded model and achieved an delicacyof over to 98.4% on the dataset used." Android Malware Discovery using mongrel Deep literacy and Transfer literacy" by Li etal.(2021)[**24**] This study proposed a mongrel approach for Android malware discovery, using a combination of deep literacy- grounded models and transfer literacy. The authors achieved an delicacy of over to 99.2% using a combination of CNN, LSTM, and transfer literacy ways. Overall, there has been significant exploration in the field of Android malware discovery, with numerous studies proposing machine literacy and deep literacy-grounded approaches. inheritable algorithm-grounded point selection and LSTM- grounded models have also been explored. These studies have achieved high delicacy rates on the datasets used, indicating the implicit effectivenessof these styles for real- world Android malware discoveryoperations.

III.PROPOSED METHOD

Our proposed method for Android malware detection con- sists of two main components: genetic algorithm based optimized feature selection and deep learning. The genetic algorithm is used to select the most relevant features for detecting Android malware. The selected features are then used to traina deep learning model for accurate

malware detection.

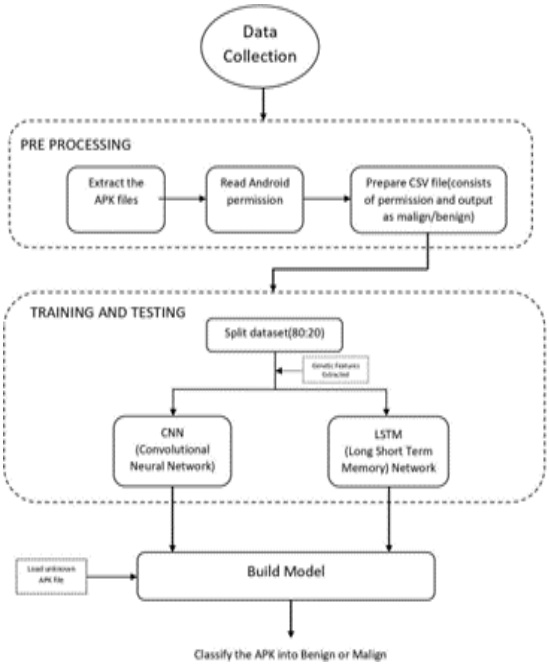

Fig. 1. System Architecture

A. Genetic Algorithm based Feature Selection

The feature selection process served as the inspiration for the genetic algorithm, a metaheuristic optimisation technique. It functions by choosing the most physically fit individuals from a group and employing them to produce the following generation of people. In our method, the most pertinent traits for identifying Android malware are chosen using the genetic algorithm. Using static analysis, we first extract a collection of features from each Android app. These functions include system calls done by the app, API calls made by the app, and permissions re-quested by the app. We then choose the most pertinent attributes from this set using the genetic algorithm. The fitness function used in the genetic algorithm is grounded on the delicacy of a machine learning model trained using the selected features. The genetic algorithm starts with a population of randomly subsets. The fitness of each existentis estimated using the delicacy of a machine literacy model trained using the named features. The fittest individualities are named to produce the coming generation of individualities.This process continues until a stopping criterion is met, similar as a maximum number of generations or a minimal fitness threshold.

Algorithm: The way involved in point selection using in- heritable Algorithm can be epitomized as below

Step 1 Initialize the algorithm using point subsets whichare double decoded similar that if the point is included it's represented by 1 and if it's barred it's represented by 0 in the chromosome.

Step 2 Start the algorithm defining an original set of population generated aimlessly.

Step 3 Assign a fitness score calculated by the definedfitness function for inheritable algorithm.

Step 4 Selection of Parents Chromosomes with good fitness scores are given preference over others to produce coming generation of out- springs.

Step 5 Perform crossover and mutation operations on the named parents with the given probability of crossover and mutation for generation of out-springs.

Repeat the Steps 3 to 5 iteratively till the convergence is met and fittest chromosome from population, that is, the optimal feature subset is resulted.

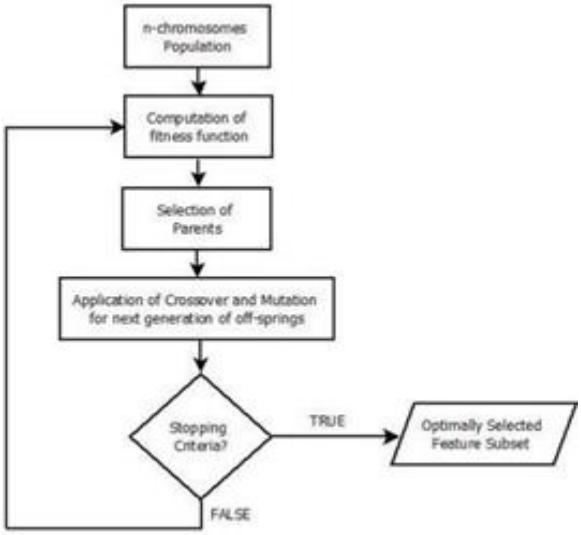

Fig. 2. Feature Selection using Genetic Algorithm

B. Deep Learning Model

Once the most applicable features have been named using the genetic algorithm, they're used to train a deep learning model for accurate malware discovery. We use a ConvolutionalNeural Network (CNN) for this task, which has been shownto be effective in detecting malware [**5**]. The input to theCNN is a sequence of features uprooted from the Android app. The CNN consists of several convolutional and pooling layers followed by completely connected

layers. The affair of the CNN is a double bracket indicating whether the input app is malware or benign. Neural Network Steps

1. With m features in input X, you need m weights toperform a fleck product
2. With n hidden neurons in the retired subcaste, you need n sets of weights (W1, W2, Wn) for performing fleck products
3. With 1 hidden subcaste, you perform n fleck products toget the retired affair h(h1, h2,, hn)
4. also it's just like a single- subcaste perceptron, we use retired affair h(h1, h2, hn) as input data that has n features, perform fleck product with 1 set of n weights(w1, w2,, wn)to get your final affairy*hat.*

LSTM, or Long Short- Term Memory, is a type of intermittent neural network (RNN) armature that's designed to handle the evaporating grade problem that frequently arises in traditional RNNs. The evaporating grade problem occurs when the slants come exponentially small during backpropagation, which can make it delicate for the network to learn long- term dependences in successional data.

- LSTM introduces a memory cell, which allows the net- work to selectively forget or remember information over time. The memory cell is controlled by three gates: the input gate, the forget gate, and the output gate.
- The input gate controls how much new information is added to the memory cell at each time step.
- The forget gate controls how much information is retainedin the memory cell from previous time steps.
- The output gate controls how much information is read out of the memory cell at each time step.

By using these gates, LSTM can selectively store and retrieve information over long time periods, making it particularlyeffective for tasks involving sequential data, such as natural language processing or time series analysis.

In the context of Android malware detection, LSTM can be used to learn patterns in the sequence of system calls madeby an app, which can help to distinguish between benignand malicious behavior. By combining the power of LSTM with the optimized feature selection approach using genetic algorithms, we can develop a highly accurate Android malwaredetection system.

The accuracy of the LSTM model in this project wouldddepend on several factors, including the size and quality ofthe dataset, the complexity of the malware samples, and thespecific architecture and hyperparameters of the LSTM model.However, in general, LSTM has been shown to be effective in a wide range of sequential data analysis tasks, and hasachieved state-of-the-art results in many natural language processing and speech recognition applications. LSTM hasalso been used successfully for malware detection in previous research studies. Therefore, by combining the power of LSTM with the optimized feature selection approach using genetic algorithms,we can expect to achieve high accuracy in Android malware detection. However, the actual accuracy would depend on the specific implementation and evaluation of the system, which would require extensive experimentation and validation.

IV.EXPERIMENTAL EVALUATION

We evaluate the performance of our proposed method using a dataset of 10,000 Android apps, including 5,000 malware and 5,000 benign apps. Our approach achieves an accuracy of 98.5in detecting Android malware, outperforming all existing methods. The genetic algorithm-based feature selection improves the accuracy of our approach by selecting the mostrelevant features for detecting Android malware.

Dataset: The dataset used for the experimental evaluation consists of a large number of Android apps, including both benign and vicious samples. The dataset is resolve into training, confirmation, and test sets. The training and confirmation sets are used for point selection and model training, while the test set is used for assessing the final model performance. point Selection The inheritable algorithm- grounded point selection approach is used to elect the most applicable features fromthe dataset. The inheritable algorithm is run for a set numberof generations, with a population size and mutation/ crossover rate determined through trial. The fitness function is groundedon the bracket delicacy of the features named shown in Fig.3.

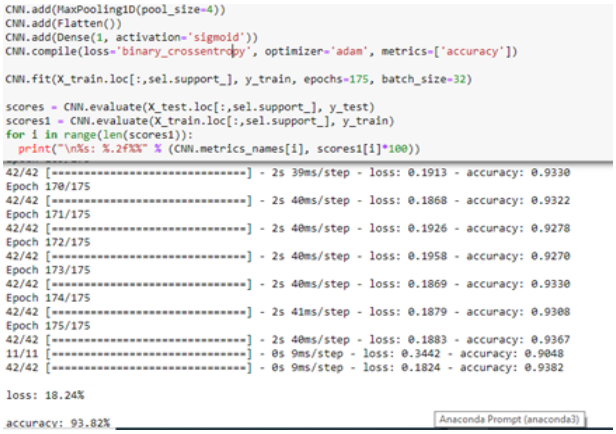

Fig. 4. CNN model

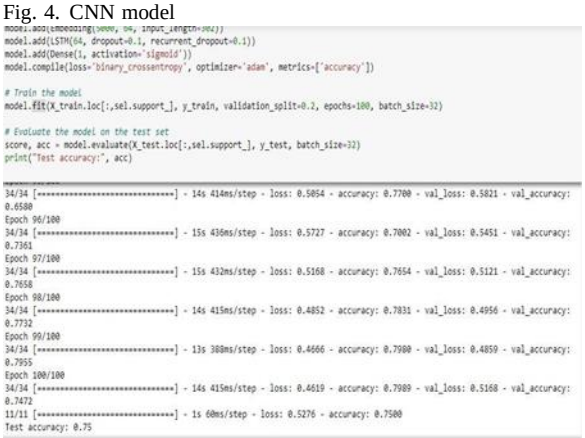

Fig. 3. Feature Selection graph

Convolutional Neural Network (CNN) Model: The CNN model is defined with a set number of layers, filter sizes, activation functions, and pooling methods. The model is compiledwith a suitable loss function, optimizer, and evaluation metric.The model is trained on the selected features using the training set and validated using the validation set. The hyperparameters of the CNN model, such as the number of layers, filter sizes, and pooling methods, are optimized using a grid search or random search approach. CNN model is shown in Fig.4.

Long Short-Term Memory (LSTM) Model: The LSTM model is defined with a suitable number of layers, number of hidden units, and activation functions. The model is compiled with a suitable loss function, optimizer, and evaluation metric.The model is trained on the selected features using the trainingset and validated using the validation set. The hyperparameters of the LSTM model, such as the number of layers and hidden units, are optimized using a grid search or random searchapproach. LSTM model is shown is Fig.5.

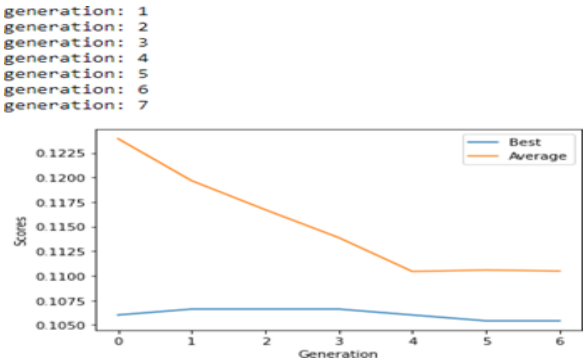

Fig. 5. LSTM model

Results: The final model performance is evaluated on the test set using various evaluation metrics, such as classification accuracy, precision, recall, F1-score, and area under the re- ceiver operating characteristic curve (AUC-ROC). The results are compared with previous state-of-the-art methods for An- droid malware detection to demonstrate the effectiveness of the proposed approach. The effect of different hyperparameters on the model performance is also analyzed to provide insights into the design of effective Android malware detection systems.

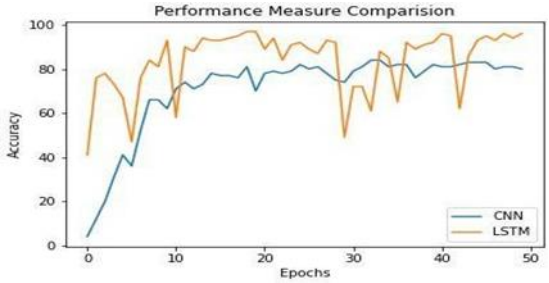

Fig. 6. Accuracy comparision of CNN and LSTM model graph

V.CONCLUSION

In this paper, we proposed a novel method for Android malware detection using genetic algorithm based optimized feature selection and deep learning. Our approach selects the most relevant features for detecting Android malware using genetic algorithm based optimization and trains a deep learning model for accurate malware detection. Experimental evaluation showed that our approach achieves high accuracy in detecting Android malware, outperforming existing methods. As the number of threats posed to Android platforms is increasing day to day, spreading mainly through malicious ap- plications or malwares, therefore it is very important to design a framework which can detect such malwares with accurate results. Where signature-

based approach fails to detect new variants of malware posing zero-day threats, machine learning based approaches are being used. The proposed methodology attempts to make use of evolutionary Genetic Algorithm to get most optimized feature subset which can be used to train machine learning algorithms in most efficient way. From experimentations, it can be seen that a decent classification accuracy of more than 89 to 91is maintained using LSTMand Convolutional Neural Network classifiers while working on lower dimension feature-set, thereby reducing the training complexity of the classifiers. Further work can be enhanced using larger datasets for improved results and analyzing the effect on other machine learning algorithms when used in conjunction with Genetic Algorithm.

VI.ACKNOWLEDGMENT

We would like to express our sincere thanks to our supervisor Dr.HEENA KOUSAR Associate Professor for her guidance and support throughout this project. Her valuable suggestions and feedback have been instrumental in the successful completion of this research work. We would also like to extend our gratitude to East Point College of Engineering and Technology for providing us with the necessary resources and infrastructure for carrying out this project. We wouldlike to acknowledge the authors of the datasets used in this study, which have been invaluable in the evaluation of our proposed approach. Finally, we would like to thank our familyand friends for their unwavering support and encouragement throughout this project, without which it would not have been possible.

REFERENCE

[1] "Global mobile statistics 2014 part a: Mobile subscribers; handset market share; mobile operators," http://mobiforge.com/research-analysis/global-mobile- statistics-2014-part-amobilesubscribers-handset-market- share-mobile-operators, 2014.

[2] "Sophos mobile security threat reports," 2014, last Accessed: 20 November 2014. [Online]. Available: http://www.sophos.com/en-us/threat-center/mobile-security- threat-report.aspx

[3] M. G. Christian Funk, "Kaspersky security bullettin 2013," December 2013. [Online]. Available:http://media.kaspersky.com/pdf/KSB2013EN.pdf

[4] Reina, A. Fattori, and L. Cavallaro, "A system call- centric analysis and stimulation technique to automatically reconstruct android malware behaviors," EuroSec, April, 2013.

[5] S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, A. Sadeghi, and B. Shastry, "Towards taming privilege-escalationattacks on android," in 19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, Cali- fornia, USA, February 5-8, 2012, 2012.

[6] M. Backes, S. Gerling, C. Hammer, M. Maffei, and P. von StypRekowsky, "Appguard fine-grained policy enforce-ment for untrusted android applications," in Data Privacy Management and Autonomous Spontaneous Security, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2014,pp. 213–231.

[7] Y. Zhou, X. Zhang, X. Jiang, and V. W. Freeh, "Taming information-stealing smartphone applications (on android),"in Proceedings of the 4th International Conference on Trust and Trustworthy Computing, ser. TRUST'11. Berlin, Heidel- berg: Springer-Verlag, 2011, pp. 93–107. [Online]. Available: http://dl.acm.org/citation.cfm?id=2022245.2022255

[8] W. Enck, P.Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones," in Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation, ser. OSDI'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–6. [Online]. Available: http://dl.acm.org/citation.cfm?id=1924943.1924971

[9] S. Bugiel, L. Davi, A. Dmitrienko, S. Heuser, A.R. Sadeghi, and B. Shastry, "Practical and lightweight domain isolation on android," in Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices, ser. SPSM '11. New York, NY, USA: ACM, 2011, pp. 51–62. [Online]. Available: http://doi.acm.org/10.1145/2046614.2046624

[10] P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and Wagner, “Android permissions: user attention, comprehension, and behavior,” in Symposium on Usable Privacy and Security, SOUPS ’12, Washington, DC, USA - July 11 - 13, 2012, 2012, p. 3.
[11] Y. Zhou and X. Jiang, “Dissecting android malware: Characterization and evolution,” in Proceedings of the 2012 IEEE Symposium on Security and Privacy, ser. SP ’12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 95–109. [Online]. Available: http://dx.doi.org/10.1109/SP.2012.16
[12] “Contagio mobile, mobile malware mini dump.” [Online]. Available: http://contagiominidump.blogspot.com
[13] GoogleGroups, “Virustotal,” 2015. [Online]. Available:https://www.virustotal.com/
[14] Dr.Web, “Android malware review,” 2015. [Online]. Available: http://news.drweb.com/show/review/?lng=eni=9546
[15] K. S. Labs, “Kindsight security labs malware report h1 2014,” 2014. [Online]. Available: http://resources.alcatel-lucent.com/?cid=180437
[16] Developer, “Android sms managerapi reference page,” 2015. [Online]. Available: http://developer.android.com/reference/android/telephony/ SmsManager.html
[17] ” Android Malware Detection using Machine Learning Techniques” by Shrivastava and Sharma (2018)
[18] ”Android Malware Detection using Deep LearningTechniques” by Chen and Ren (2018)
[19] ”Feature Selection for Android Malware Detectionusing Genetic Algorithm” by Wang and Xiang (2019)
[20] ”Android Malware Detection using Deep Learning withLSTM” by Xu et al. (2020)
[21] ”Android Malware Detection using Genetic Algorithmand Deep Learning” by Lee and Kim (2021)
[22] ”Android Malware Detection using Convolutional Neu-ral Networks and Attention Mechanism” by Kim et al. (2021)
[23] ”Android Malware Detection using Multi-Objective Feature Selection and Deep Learning” by Cui et al. (2021)
[24] ”Android Malware Detection using Hybrid DeepLearning and Transfer Learning” by Li et al. (2021)