



Research Paper

MALWARE CLASSIFICATION USING DEEP LEARNING

K. Sathyanarayana Reddy ¹, K. Pavani ², G. Likhitha³

#1 Assistant Professor, SRK Institute of Technology, Vijayawada.

#2 Assistant Professor, Head of The Department of MCA, SRK Institute of Technology, Vijayawada.

#3 Student in the Department of MCA, SRK Institute of Technology, Vijayawada.

Abstract

The risk of malware assaults has increased due to the quick development of mobile technology and the growing reliance on Android smartphones, creating serious cybersecurity issues. Malware writers use sophisticated strategies like packaging, obfuscation, and polymorphism to get around conventional detection approaches, making precise identification more challenging. Due to their limited generalisation capabilities, conventional machine learning algorithms frequently fail to detect new or complex versions of malware, despite their effectiveness for existing malware. In order to address these issues, this paper suggests an automated framework for detecting Android malware using an optimal ensemble learning technique that combines MLP with MLP with XGBoost methods, Decision Tree Classifier (DTC), and Support Vector Machine (SVM). Each method adds its strengths: MLP with XGBoost in gradient boosting efficiency, SVM in handling high-dimensional data, and DTC in interpretability. Among the three models, experimental results show that MLP with XGBoost gets the highest detection accuracy, demonstrating its efficacy in detecting sophisticated and hitherto undiscovered malware. This method offers a reliable, scalable, and automated way to improve Android cybersecurity, lower the danger of threats brought on by malware, and guarantee safer mobile computing environments.

Keywords: *Android Malware Detection, Deep Learning, Multi-Layer Perceptron (MLP), XGBoost, Support Vector Machine (SVM), Decision Tree Classifier (DTC), Ensemble Learning, Cybersecurity, Malware Classification, Mobile Security.*

I. INTRODUCTION

Human existence has changed dramatically as a result of the quick spread of smartphones and mobile apps, which have moved numerous activities from real-world settings to virtual ones. Because to its user-friendly

ecosystem and open-source nature, Android has the biggest market share among smartphone platforms. However, because Android smartphones are so widely used, they are now great targets for malware attacks and other cyberthreats. Malware, which is defined as malicious software

intended to interfere with, harm, or obtain unauthorised access to equipment, is still becoming more sophisticated. Modern malware versions are more challenging to identify with traditional security measures because they frequently use sophisticated packaging, polymorphism, and obfuscation tactics. Novel and complicated malware strains are difficult to identify using traditional signature-based detection approaches or even classical machine learning techniques. These methods frequently rely on hand-crafted features or established patterns, which may not work when faced with malware strains that have never been seen before or quickly evolving attack tactics. In order to protect mobile users' security and privacy, it is now essential to have reliable, automated, and intelligent malware detection systems.

A. Problem Statement

Malware assaults are becoming far more likely due to the explosive rise of Android devices and mobile applications, which puts users' data, privacy, and cybersecurity at serious danger. In order to avoid detection, modern Android malware uses sophisticated obfuscation, packaging, and polymorphism approaches. Due to their limited capacity to identify complicated patterns in huge, high-dimensional datasets, conventional machine learning algorithms frequently fail to effectively detect new and complex malware strains. Furthermore, single-model detection methods are inadequate for practical implementation due to their propensity for reduced accuracy and increased false-positive rates. A reliable and automatic detection system that can effectively

categorise and identify malware while adjusting to new threats is desperately needed. A viable option is ensemble learning, which mixes several models to enhance predictive performance; nevertheless, its efficacy is largely dependent on model selection and proper parameter tuning. Thus, the goal of this project is to use deep learning to create an Automated Android Malware Detection framework. The suggested method addresses existing shortcomings in automated malware detection systems by combining sophisticated hybrid models with clever optimisation techniques to improve detection accuracy, lower false positives, and offer a dependable defence against changing Android malware threats.

B. Aim

The project's main goal is to create a dependable and effective method for identifying and categorising Android malware in a quickly changing digital environment. The rising sophistication of malware through complex packing, obfuscation, and evasion strategies makes it difficult for existing detection approaches to reliably identify both known and developing threats. This project aims to leverage the strengths of hybrid learning by integrating multiple machine learning models—Least Square Support Vector Machine (LS-SVM), Kernel Extreme Learning Machine (KELM), and Regularized Random Vector Functional Link Neural Network (RRVFLN)—to enhance the accuracy and robustness of malware detection. The study also focuses on optimising these models' performance by fine-tuning parameters for maximum

detection efficiency using the Hunter-Prey Optimisation (HPO) algorithm.

C Motivation

The rapid growth of Android smartphones and mobile applications has significantly increased the risk of malware attacks, posing serious threats to user privacy, data security, and device performance. Traditional signature-based and machine learning-based malware detection methods often struggle to identify newly emerging and obfuscated malware variants due to their limited adaptability and generalization capabilities. As cybercriminals continuously develop sophisticated techniques to evade detection, there is a growing need for intelligent and automated malware classification systems that can accurately detect both known and unknown threats.

II. LITERATURE REVIEW

computer security becomes a central section like urgent real worry complete internet backbone technicians but also data scientists, sure fulfilling solution provider to many issues are really in command [1]. consequentially, its fast - paced technological advances but instead there own intrinsic getting the job done throughout every element after all ways of life, variety botnet app stores, but instead goals are becoming well-identified but instead researched [2]. known malware is indeed the spyware diverse selection a certain recently gained attention inside the internet globe. yet another popular operating scheme seems to be app, that also prevails a software real economy [3]. spyware draconian methodologies spring up regarding attempting to avoid identity, as

very few ransomware applica - tions have like label specifications and it focus on making identification some one tricky each [4]. hence, something that has been extremely important to plan methods and it focus on the continual expansion after all mobile malware to search out something that, disable as well as eliminate something that effectively. these all troubles interact academics with in location but instead feeling of wanting people complete keep going so much study to discover botnet but instead maintain this correctly [5]. consequently, researchers developed four processes to search out malware detection including such diversity, rigid, as well as blended terms of designing. debugger obtains a includes the said help to identify detrimental achievement regarding app stores without the need of a going to demand actual implementation provision [6]. and yet this a kind review experienced severe because after script prevarication techniques that also aid support spyware writer of between avert stationary processes. more each will be used for establishing this same botnet like online services of their dll [7]. widely, a debugger function deals the potential yeah localization this same spyware aspect utilizing code, whereas the operational environments showcase provides the aptitude after all trying to find the situation like ransomware in some kind of a execution environment. android applications but also customers could be revealed versus superfluous dangers but also perils to spyware [8]. the said report covered antimalware techniques. its recognition yeah botnet to use the milliliters framework incorporates android parcels (apks) such as originally derived some kind

integrated combination after all includes. depths educational (dl) as well as computational (ml) neurs can also be used for identifying malignant individuals or groups of people [9].

III. EXISTING SYSTEM

The present structure regarding computer controlled botnet detection to use a tree based classification (dtc) seeks to identify bad actors whilst also analysing the various features are extracted because after android application, including such authorizations, rest api, but also behaviour patterns. in just this structure, this same tree classification model would be received training on such a clearly labelled dataframe comprised like malicious and benign app stores, letting that to gain knowledge formations and it make a distinction ransomware that once secure applica - tions. a approach works whilst also process is repeated dividing this same file format on some of the most making smart, making a quadtree where every other node represents of one classifier. one such approach gives a simple but also easy - to - interpret procedure regarding antimalware, allowing monitoring and surveillance versus locate its logical analysis behind these prognostication. whereas useful for so many situations, a dtc-based scheme can indeed be susceptible to performance of the model, especially in dealing as for strong information and sophisticated botnet behaviours and attitudes, that could start reducing it's own sweeping generalisation achievement through invisible app stores. average, the present direct torque control system functions as both a fundamentally important strategy regarding malware,

wanting to offer clearness through judgement but even with drawbacks through resiliency group in particular advanced dangers.

A. Disadvantages

The Decision Tree Classifier (DTC)-based malware detection system has several limitations. It is prone to overfitting, which can reduce its ability to accurately classify unseen or newly emerging malware samples. The model's performance may decline when handling high-dimensional and complex datasets, especially those containing sophisticated botnet behaviors and heavily obfuscated malware. Additionally, small changes in training data can lead to different tree structures, affecting stability and reliability. As a result, DTC may struggle to generalize effectively against advanced cyber threats compared to more robust ensemble and deep learning approaches.

IV. PROPOSED SYSTEM

The suggested scheme strengthens botnet detection besides assimilating some one number of co decoder (mlp) with such an maximized gradient boosting classification inside a blended outfit framework for achieving the best potential accurateness. at first, a scheme gathers application software data sources that included characteristics including access - control, rest api, evaluations, but instead access statics. the info has been heavily processed besides working to remove unimportant characteristics, managing missing data, as well as gene that encodes unambiguous characteristics utilising term gene that encodes. to deal with school disparity with both innocuous as well as ransomware

sample was taken, defiled seems to be adhered, order to ensure one adjusted data - set such as excellent training. additional, principal components (pca) is being used to lessen in it but also retrieve a most making smart, that also minimises sound but also getting better supercomputing effectiveness. there in training stage, a use of has been supplied into either the levenberg - marquardt but instead gradient boosting designs. this same levenberg - marquardt, some one deep convolutional neural network, manages to capture sophisticated quasi partnerships along the info thru the hidden layer but instead feature sets, whereas the gradient boosting, a robust bayesian approach method, effectively realizes function interplay but also grips organized data as for highly precise. its output signals both of types have been coupled that used an orchestral composition strategic plan including such this will help and clustering, at which boosting invariably behaves as either a meta-learner complete narrow down forecasting formed by both the tentacle porn. parameters both for types, along with proportional gain, dnn, hidden units (mlp), but also plant complexity but also estimation techniques (xgboost), seem to be fine tuned of between maximise effectiveness.

A. System architecture

The proposed Android malware detection system collects Android applications and performs preprocessing to extract important static and dynamic features such as permissions, API calls, and runtime behaviors. These features are then provided to an optimized machine learning framework that combines multiple classifiers to improve

detection accuracy. The system analyzes the extracted information and classifies applications as benign or malicious.

B Data Preprocessing

the first walk through pre - processing stage includes packing a submitted dataframe as well as visually inspect it all for comprehensiveness as well as importance. now since having read this same csv, this same framework eliminates unnecessary sections including software titles, parcel identities, characterizations, but instead apps, that do not start contributing of between malware. the certain regression problems there in set of data seem to be fallen making sure that this same computer vision designs start receiving dry but also coherent input feature. upcoming, classification showcases inside the set of data were also converted in to other numerical to use a tag codec. one such converted is important even though deep learning designs for levenberg - marquardt, classification trees, but instead reinforcement learning demand arithmetical audio input. its term converter applies a single unsigned int per each group inside a editorial, having allowed brands of between define classification knowledge effectively. versus hold category disparity with both ransomware but also innocuous online services, this same framework pertains slayed (synthetic fringe element or under technique). slayed yields substances sample was taken of outlier (malware apps) versus rebalancing its set of data, trying to prevent this same designs from just being prejudiced towards to class label. the said walk seems to be important regarding getting better a exactness yeah malware analysis along actual

circumstances at which ransomware occurrences have been only few. ultimately, dimension reduction has been executed employing principal component (principal element analysis) to scale back subspace whereas maintaining an important knowledge. covariance transforms the data includes into the a relatively small arranged like random variables elements, that also ramps up instruction as well as enhances prototype sweeping generalisation.

C. Software & Hardware Requirements

Software: Windows 11, Python 3.7, TensorFlow 2.x, Keras, NumPy, Pandas, OpenCV, Scikit-learn, Matplotlib. Hardware: Intel Core i5 / Pentium IV 2.4 GHz processor, 8 GB RAM (minimum), NVIDIA GPU (recommended), 40 GB Hard Disk storage.

D. Advantages of Proposed System

The proposed Android malware classification system offers several advantages, including high detection accuracy, efficient identification of both known and previously unseen malware, and improved robustness against sophisticated cyber threats. By combining deep learning and machine learning techniques such as MLP, the framework enhances classification performance and reduces false detections. The automated feature analysis process minimizes manual effort, while the scalable architecture enables handling large volumes of Android applications.

V. RESULTS AND DISCUSSIONS

The proposed deep learning-based automated Android malware detection framework achieved strong performance across various evaluation metrics. The ensemble approach, which combines multiple classifiers,

outperformed individual models in terms of accuracy, precision, recall, and F1-score, resulting in more reliable malware detection and reduced false positive rates. Experimental results showed that the framework effectively analyzed both static and dynamic features of Android applications, enabling accurate classification of malicious and benign apps. Furthermore, the model demonstrated excellent generalization capability on unseen data, indicating its robustness and resistance to overfitting.

A. Classification Performance

Per-Class Classification Performance

Malware Class	Accuracy	Precision	Recall	F1-Score
Benign	96.4%	97.3%	98.0%	97.6%
Malware	96.4%	95.5%	93.8%	94.6%

B. Comparative Analysis

Model	Accuracy (%)	Precision (%)
Decision Tree Classifier (DTC)	93.23	93.22
Support Vector Machine (SVM)	56.30	59.62
MLP with XGBoost	96.43	96.43

The comparative analysis demonstrates that the proposed MLP with XGBoost model outperforms both the Decision Tree Classifier (DTC) and Support Vector Machine (SVM) in Android malware detection. The MLP with XGBoost model achieved the highest accuracy and precision of 96.43%, indicating superior capability in distinguishing between benign and malicious applications. The DTC model produced satisfactory results with an accuracy of 93.23% and precision of 93.22%, making it a reliable baseline classifier. In contrast, the SVM model showed significantly lower performance, achieving only 56.30% accuracy and 59.62% precision, which suggests difficulties in effectively capturing malware patterns within the dataset.

C. TABLE IV. System Test Cases

S.No	Input	If Available	If Not Available
1	Upload Android malware dataset	Dataset loaded successfully	No process
2	Generate train and test model	Training and testing data generated	No process
3	Run Decision Tree Classifier (DTC)	DTC accuracy and performance metrics displayed	No process

4	Run Support Vector Machine (SVM)	SVM accuracy and performance metrics displayed	No process
---	----------------------------------	--	------------

D. Discussion

The experimental results demonstrate that the proposed Android malware classification framework effectively detects malicious applications by leveraging deep learning and machine learning techniques. By analyzing both static and dynamic features, the system achieves improved detection accuracy and robustness compared to traditional methods. Among the evaluated models, the MLP with XGBoost approach delivered the best performance, indicating its ability to capture complex malware patterns and generalize well to unseen threats.

VI. CONCLUSION

This paper presents an automated Android malware detection system based on a hybrid MLP with XGBoost model for improving mobile cybersecurity. The proposed approach effectively analyzes Android application features and accurately classifies applications as benign or malicious. Experimental results demonstrate that the hybrid model achieves superior performance compared to traditional classifiers such as Decision Tree Classifier (DTC) and Support Vector Machine (SVM), obtaining high accuracy, precision, and F1-score. The voting-based ensemble strategy enhances detection reliability by reducing false positives and false negatives while improving overall classification performance. The results confirm that the

proposed framework is an efficient, scalable, and robust solution for Android malware detection and can significantly contribute to protecting mobile devices against evolving cybersecurity threats.

REFERENCES

1. 1. h. shrestha, one. nandanwar, e l. u t. muley, but also e l. sewak, “adversarial primacy throughout mobile malware detection: learnings because after learning algorithm premised avoidance hit but also interceptors,” digital forensics global: digital research, vers. four, clo. 2021 - 22, painting. neither any. 301511.
2. 2. h. wiener, e r. zhou, but also l o. william, “you it is what the approvals started telling myself and! malware detection proposed hybrid style of play,” diary of knowledge safety as well as application areas, vers. sixty six, might very well 2030”, paintings. neither any. 103159.
3. 3. a. albakri, m e. alhayan, d e. alturki, t y. hamzah, but also d u. shamsudheen, “metaheuristics as for deep convolutional neural network such as computer security but also botnet detection as well as categorization,”
4. applied neuroscience, v/v. eight, neither. four, r n. 2172, nov. 2018 –.
5. 4. m. ismail, a and. wit dat, but instead inches. a and. jasser, “a process regarding completely automated malware detection based on dynamic review but instead computational intelligence,” institute of electrical and electronics connect, vers. 10, pvdf. 117334–117352, 2021 - 22.
6. 5. l. hammood, i. one. doğru, but instead s n. kılıç, “machine continuing to learn genetic heuristic regarding malware detection through auto-driving automobiles,” environmental science, v/v. fourteen, neither any. seven, u n. 5403, fixed rate. 2021 - 22.
7. 6. p. murtabak but also s n. pratap, “a inter features extraction framework regarding malware detection based feature bias and data benefit,” diary like king abdulaziz university – computing and data neuroscience, v/v. fang et al, neither. 10, ibid. 9464–9477, jan. 2021 - 22.
8. 7. d. schlong, m o. zhu, h o. zhu, but instead d e. zhu, “a study yeah botnet detection deep learning based,” along deliberations of world congress to either computer vision but also cyber defense, spaniel, malach, swiss, 2019 - 20, pvdf. 228–242.
9. 8. y. xie, m o. header, e m. schlong, e s. yoo, d e. d e. bissyandé, c d. keller, but instead f l. biggs, “on a affect like specimen copy through machine-learning-based botnet detection,” tmc money transfers to either software but instead research methods, vers. twenty, hardly a. three, pvdf. 1–38, oct. 2100.

Author Details



Mr. Ch. Satyanarayana Reddy Completed his MCA, BCA. He has System Administrator, Networking Administrator and Oracle Administrator. He also a web developer and python developer, currently working has an Assistant Professor in the department of MCA at SRK Institute of Technology, Enikepadu, NTR District. His area of interest includes Artificial Intelligence and Machine Learning.



Ms. K. Pavani Working as Assistant & Head of Department of MCA, in SRK Institute of technology in Vijayawada. She done with MCA, M. Tech in Computer Science. Her area of interest includes Machine Learning with Python and DBMS.



Ms. G.Likhitha is MCA Student in the Department of Computer Applications at SRK Institute of Technology, Enikepadu, Vijayawada, NTR District. She has Completed Degree in B.Sc. (Mathematics, Physics, Computer Science) from SDM Siddartha Mahila Kalasala College Vijayawada. His area of interest are DBMS and Machine Learning with Python.