Automated Android Malware Detection Using Optimal Ensemble Learning Approach for Cyber security

T.Sundararajulu[1],
M.Lakshman[2],K.Omprakash[2], M.Bhavya Sri[2], P.Eliyaz Khan[2]

[1] *Assistant Professor, Department of Computer Science & Information Technology, Siddharth Institute of Engineering & Technology, Andhra Pradesh, India*
[2] *Research Scholar, Department of Computer Science & Information Technology, Siddharth Institute of Engineering & Technology, Andhra Pradesh, India*

ABSTRACT

Current technological advancement in computer systems has transformed the lives of humans from real to virtual environments. Malware is unnecessary software that is often utilized to launch cyberattacks. Malware variants are still evolving by using advanced packing and obfuscation methods. These approaches make malware classification and detection more challenging. New techniques that are different from conventional systems should be utilized for effectively combating new malware variants. Machine learning (ML) methods are ineffective in identifying all complex and new malware variants. The deep learning (DL) method can be a promising solution to detect all malware variants. This project presents an Automated Android Malware Detection using Optimal Ensemble Learning Approach for Cybersecurity (AAMDOELAC) technique. The major aim of the AAMD-OELAC technique lies in the automated classification and identification of Android malware. To achieve this, the AAMD-OELAC technique performs data preprocessing at the preliminary stage. For the Android malware detection process, the AAMD-OELAC technique follows an ensemble learning process using three ML models, namely Least Square Support Vector Machine (LS-SVM), kernel extreme learning machine (KELM), and Regularized random vector functional link neural network (RRVFLN). Finally, the hunter-prey optimization (HPO) approach is exploited for the optimal parameter tuning of the three DL models, and it helps accomplish improved malware detection results. To denote the supremacy of the AAMD-OELAC method, a comprehensive experimental analysis is conducted.

Keyword: - *Malware, Android, optimal, Cyber security*

1. INTRODUCTION

In our quest to fortify the static detection of Android malware, we introduce a novel subset of features meticulously curated from diverse categories. These seven additional feature sets are carefully selected for their potency in discerning between benign and malicious Android applications. Our feature selection spans permissions, API call sequences, manifest attributes, resources, code characteristics, behavioral patterns, and metadata. Each feature set contributes unique insights into the application's behavior, empowering our detection model to scrutinize every facet of an app's structure and operation.

To evaluate the robustness of our approach, we employ a vast dataset comprising over 500,000 Android applications, encompassing both benign and malicious samples. Notably, our dataset boasts the largest malware sample set compared to existing methodologies, ensuring comprehensive coverage of contemporary threats. We meticulously curate this dataset to include samples collected over recent years, imbuing our model with time-awareness and relevancy to the latest malware trends. Rigorous cross-validation techniques and performance metrics such as precision, recall, F1-score, and ROC-AUC are employed to assess the stability and efficacy of our approach.

In tandem with our feature-rich framework, we deploy six classifier models encompassing a spectrum of machine learning algorithms. Decision Trees, Random Forests, Support Vector Machines, Logistic Regression, Gradient Boosting Machines, and Neural Networks are among the models harnessed to scrutinize our feature sets and discern malicious from benign applications. Additionally, we employ a Boosting ensemble learning approach, specifically AdaBoost, coupled with a Decision Tree base learner. This ensemble technique amalgamates the predictive prowess of individual models, amplifying our detection capabilities and fortifying our resilience against evolving malware threats. Through this holistic approach, we establish a formidable defense against Android malware, anchored in advanced feature engineering, cutting-edge machine learning, and ensemble learning strategies

2. LITERATURE SURVEY

[1] AUTOMATED ANDROID MALWARE DETECTION USING OPTIMAL ENSEMBLE LEARNING APPROACH FOR CYBER SECURITY

Current technological advancement in computer systems has transformed the lives of humans from real to virtual environments. Malware is unnecessary software that is often utilized to launch cyberattacks. Malware variants are still evolving by using advanced packing and obfuscation methods. These approaches make malware classification and detection more challenging. New techniques that are different from conventional systems should be utilized for effectively combating new malware variants. Machine learning (ML) methods are ineffective in identifying all complex and new malware variants. The deep learning (DL) method can be a promising solution to detect all malware variants. This paper presents an Automated Android Malware Detection using Optimal Ensemble Learning Approach for Cybersecurity (AAMDOELAC) technique. The major aim of the AAMD-OELAC technique lies in the automated classification and identification of Android malware. To achieve this, the AAMD-OELAC technique performs data preprocessing at the preliminary stage. For the Android malware detection process, the AAMD-OELAC technique follows an ensemble learning process using three ML models, namely Least Square Support Vector Machine (LS-SVM), kernel extreme learning machine (KELM), and Regularized random vector functional link neural network (RRVFLN). Finally, the hunter-prey optimization (HPO) approach is exploited for the optimal parameter tuning of the three DL models, and it helps accomplish improved malware detection results. To denote the supremacy of the AAMD-OELAC method, a comprehensive experimental analysis is conducted. The simulation results portrayed the supremacy of the AAMD-OELAC technique over other existing approaches.

[2] YOU ARE WHAT THE PERMISSIONS TOLD ME! ANDROID MALWARE DETECTION BASED ON HYBRID TACTICS

Recent years have witnessed a significant increase in the use of Android devices in many aspects of our life. However, users can download Android apps from third-party channels, which provides numerous opportunities for malware. Attackers utilize unsolicited permissions to gain access to the sensitive private intelligence of users. Since signature-based antivirus solutions no longer meet practical needs, efficient and adaptable solutions are desperately needed, especially in new variants. As a remedy, propose a hybrid Android malware detection approach that combines dynamic and static tactics. firstly, adopt static analysis inferring different permission usage patterns between malware and benign apps based on the machine-learning-based method. To classify the suspicious apps further, extract the

object reference relationships from the memory heap to construct a dynamic feature base. then present an improved state-based algorithm based on DAMBA. Experimental results on a real-world dataset of 21,708 apps show that our approach outperforms the well-known detector with 97.5% F1-measure.

[3] METAHEURISTICS WITH DEEP LEARNING MODEL FOR CYBERSECURITY AND ANDROID MALWARE DETECTION AND CLASSIFICATION

Since the development of information systems during the last decade, cybersecurity has become a critical concern for many groups, organizations, and institutions. Malware applications are among the commonly used tools and tactics for perpetrating a cyberattack on Android devices, and it is becoming a challenging task to develop novel ways of identifying them. There are various malware detection models available to strengthen the Android operating system against such attacks. These malware detectors categorize the target applications based on the patterns that exist in the features present in the Android applications. As the analytics data continue to grow, they negatively affect the Android defense mechanisms. Since large numbers of unwanted features create a performance bottleneck for the detection mechanism, feature selection techniques are found to be beneficial. This work presents a Rock Hyrax Swarm Optimization with deep learning-based Android malware detection (RHSODL-AMD) model. The technique presented includes finding the Application Programming Interfaces (API) calls and the most significant permissions, which results in effective discrimination between the good ware and malware applications. Therefore, an RHSO based feature subset selection (RHSO-FS) technique is derived to improve the classification results.

[4] A METHOD FOR AUTOMATIC ANDROID MALWARE DETECTION BASED ON STATIC ANALYSIS AND DEEP LEARNING

The computers nowadays are being replaced by the smartphones for the most of the internet users around the world, and Android is getting the most of the smartphone systems' market. This rise of the usage of smartphones generally, and the Android system specifically, leads to a strong need to effectively secure Android, as the malware developers are targeting it with sophisticated and obfuscated malware applications. Consequently, a lot of studies were performed to propose a robust method to detect and classify android malicious software (malware). Some of them were effective, some were not; with accuracy below 90%, and some of them are being outdated; using datasets that became old containing applications for old versions of Android that are rarely used today. In this paper, a new method is proposed by using static analysis and gathering as most useful features of android applications as possible, along with two new proposed features, and then passing them to a functional API deep learning model we made. This method was implemented on a new and classified android application dataset, using 14079 malware and benign samples in total, with malware samples classified into four malware classes. Two major experiments with this dataset were implemented, one for malware detection with the dataset samples categorized into two classes as just malware and benign, the second one was made for malware detection and classification, using all the five classes of the dataset.

[5] MACHINE LEARNING-BASED ADAPTIVE GENETIC ALGORITHM FOR ANDROID MALWARE DETECTION IN AUTO-DRIVING VEHICLES

The growing trend toward vehicles being connected to various unidentified devices, such as other vehicles or infrastructure, increases the possibility of external attacks on "vehicle cybersecurity (VC). Detection of intrusion is a very important part of network security for vehicles such as connected vehicles, that have open connectivity, and self-driving vehicles. Consequently, security has become an important requirement in trying to protect these vehicles as attackers have become more sophisticated in using malware that can penetrate and harm vehicle control units as technology advances. Thus, ensuring the vehicles and the network are safe is very important for the growth of the automotive industry and for people to have more faith in it. In this study, a machine learning-based detection approach using hybrid analysis-based particle swarm optimization (PSO) and an adaptive genetic algorithm (AGA) is presented for Android malware detection in auto-driving vehicles. The "CCCS-CIC-AndMal-2020" dataset containing 13 different malware categories and 9504 hybrid features was used for the experiments. In the proposed approach, firstly, feature selection is performed by applying PSO to the features in the dataset. In the next step, the performance of XGBoost and random forest (RF) machine learning classifiers is optimized using the AGA.

3. METHODOLOGY

3.1EXISTING SYSTEM

In Existing system, the primary objective is to develop a system that can automatically detect Android malware using ensemble learning techniques, which combine multiple models to improve detection accuracy. With the increasing number of Android malware threats, automated detection systems are essential for protecting users' devices and data. Existing Android malware detection systems may not be sufficiently robust or efficient, and new malware variants are constantly emerging, requiring adaptive and accurate detection mechanisms. The project proposes to use ensemble learning techniques to improve the accuracy and effectiveness of Android malware detection. Ensemble learning combines the predictions of multiple models to produce better results than any single model alone. The project likely involves collecting and analyzing datasets of known malware and benign Android applications to train and evaluate the ensemble learning models. The success of the project will be evaluated based on metrics such as detection accuracy, false positive rate, and computational efficiency of the developed ensemble learning approach compared to existing methods. Given the potential impact on user privacy and security, ethical considerations regarding data collection, use, and protection must be addressed.

3.1.1DISADVANTAGES OF EXISTING SYSTEM

- The system is not implemented Machine Learning Algorithm And Ensemble Learning.
- The system is not implemented Reverse Engineered Applications characteristics.

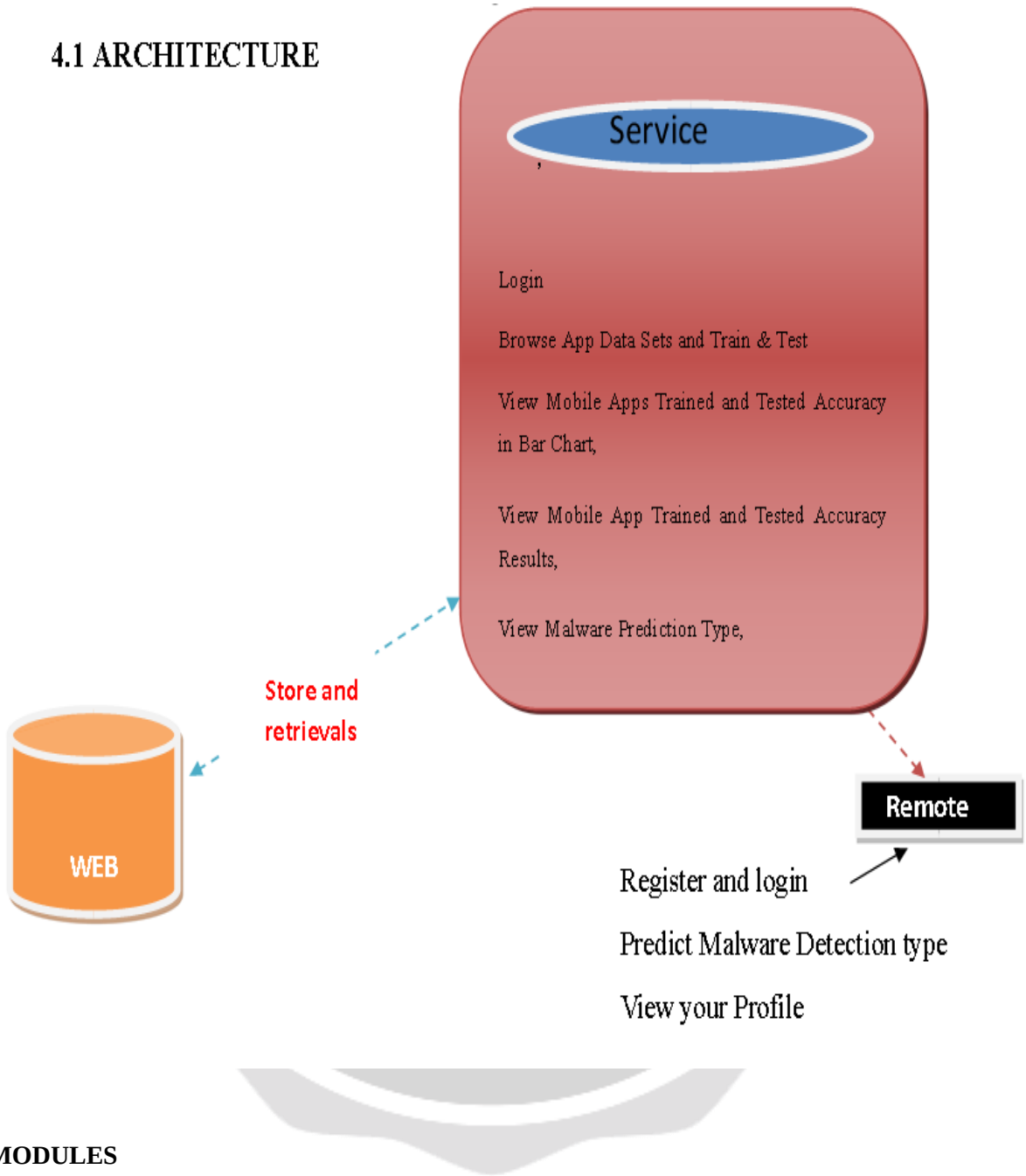
3.2 PROPOSED METHODOLOGY

- In the proposed system, present a novel subset of features for static detection of Android malware, which consists of seven additional selected feature sets that are using around features from these categories. On a collection of more than 500k benign and malicious Android applications and the highest malware sample set than any state-of-the-art approach, assess their stability.
- With the additional features, have trained six classifier models or machine learning algorithms and also implemented a Boosting ensemble learning approach (AdaBoost) with a Decision Tree based on the binary classification to enhance our prediction rate.
- Our model is trained on the latest and large time aware samples of malware collected within recent years including the latest web API level than state-of the-art approaches.

4. SYSTEM DESIGN

It is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements. Before planning, you needto understand the old system thoroughly and determine how computers can best be used in order to operate efficiently.

4.1 ARCHITECTURE



4.2 MODULES

In this Proposed System, There are two Modules. They are:

1. Service Provider
2. Remote User

4.2.1 SERVICE PROVIDER

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as Login, Browse App Data Sets and Train & Test, View Mobile Apps Trained and Tested Accuracy in Bar Chart, View Mobile App Trained and Tested Accuracy Results, View Malware Prediction

Type, View Malware Prediction Ratio, Download Predicted Data Sets, View Malware Prediction Ratio Results, View All Remote Users,Logout.

4.2.2 REMOTE USER

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like Register And Login, Predict Malware Detection Type, View Your Profile, Logout.

5. RESULTS AND PERFORMANCE

EXECUTION PROCEDURE

The Execution procedure is as follows:

1. In this research work with data with attributes are observable and then all of them are floatingdata. And there's a decision class/class variable. This data was collected from Kaggle machine learning repository.
2. In this research 70% data use for train model and 30% data use for testing purpose.
3. Logistic regressions is used as Classifier.
4. In the classification report we were able to find out the desired result
5. In this analysis the result depends on some part of this research. However, which algorithm gives the best true positive, false positive, true negative, and false negative are the best algorithmsin this analysis.

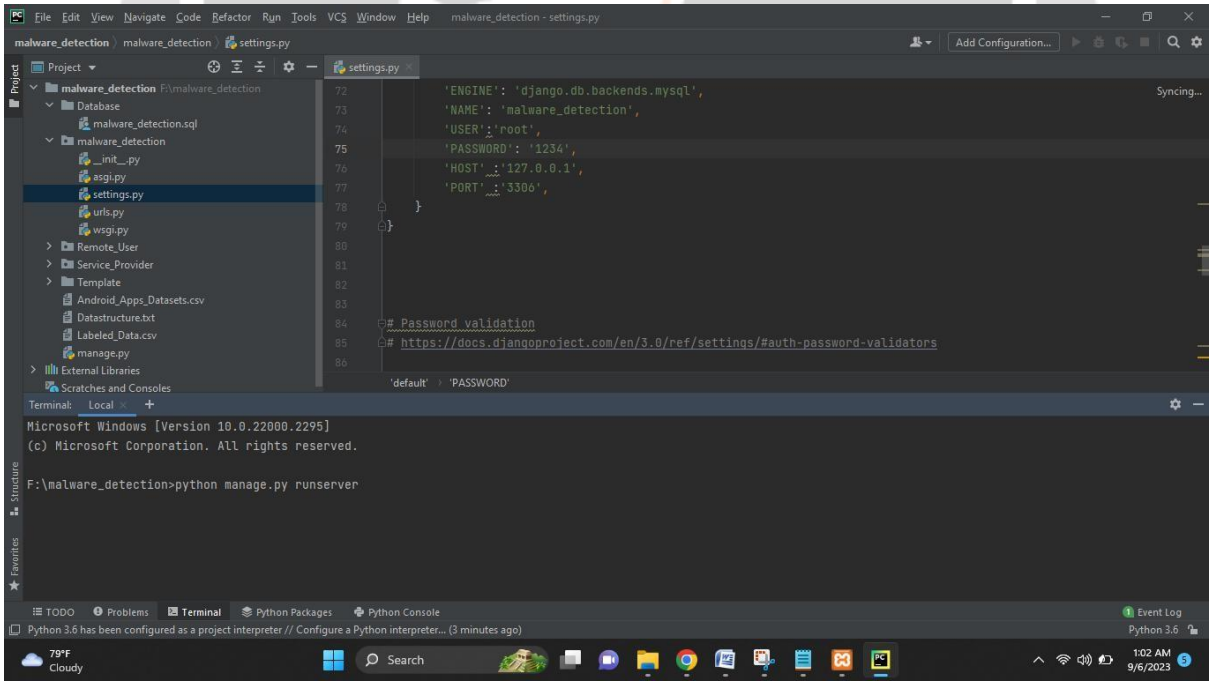


Fig . Running Program on console

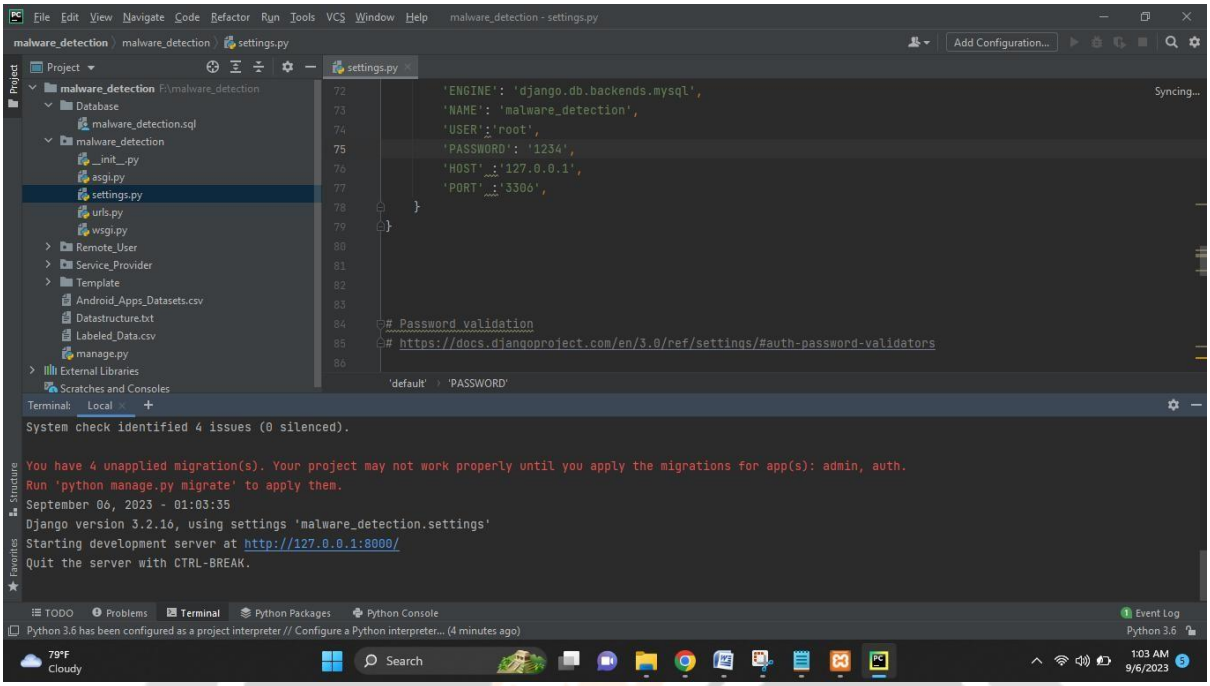


Fig. Generating Link

Fig . Service Provider Login

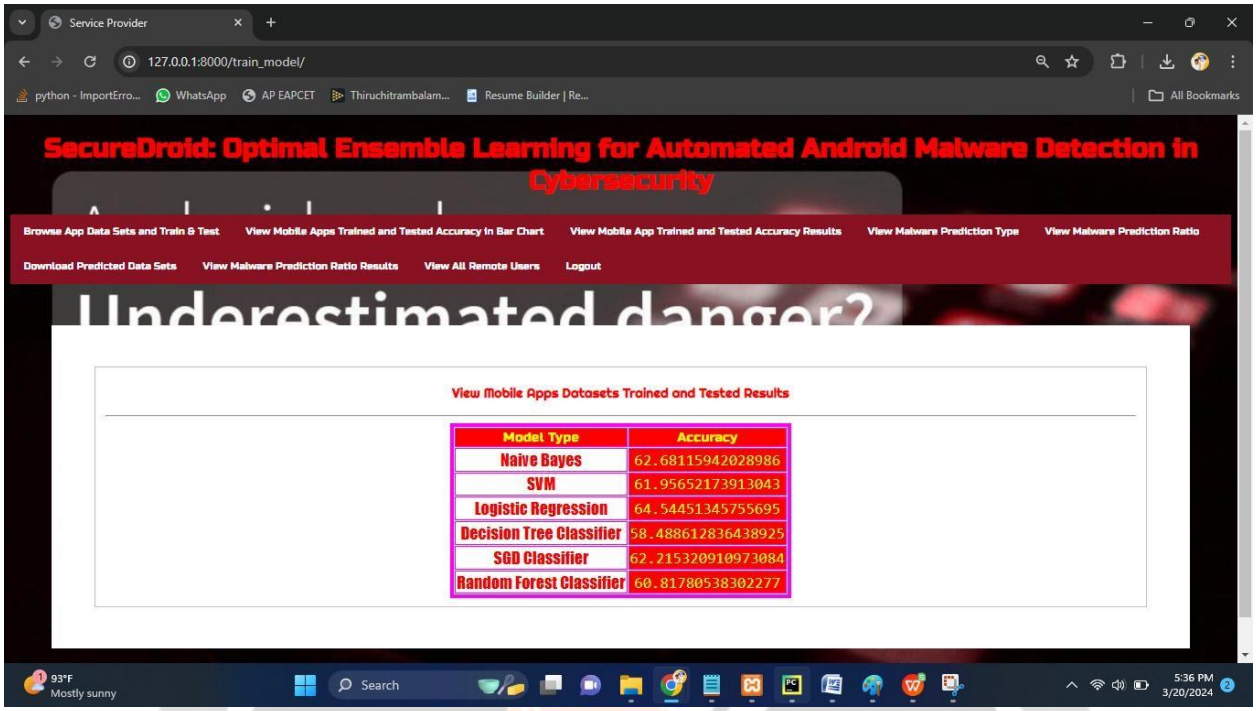


Fig.Generate Trained and Tested Accuracy

Fig. User Login

6. CONCLUSION

In this project, devised a framework that can detect malicious Android applications. The proposed technique takes into account various elements of machine learning and achieves a in identifying malicious Android applications. first define and pick functions to capture and analyze Android apps' behavior, leveraging reverse application engineering and AndroGuard to extract features into binary vectors and then use python build modules and split shuffle functions to train the model with benign and malicious datasets investigate how the whole model can be self-adaptive on the user behavior.

7. REFERENCE

[1] A. Datta, S. Buchegger, L.-H.Vu, T. Strufe, and K. Rzadca, ‘‘Decentralized online social networks,’’ in *Handbook of Social Network Technologies and Applications*. Springer, 2010, pp. 349–378.

[2] L. Jiang and X. Zhang, ‘‘BCOSN: A blockchain-based decentralized online social network,’’ *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 6, pp. 1454–1466, Dec. 2019.

[3] B. Guidi, A. Michienzi, and L. Ricci, ‘‘Steem blockchain: Mining the inner structure of the graph,’’ *IEEE Access*, vol. 8, pp. 210251–210266, 2020.

[4] W. Sherchan, S. Nepal, and C. Paris, ‘‘A survey of trust in social networks,’’ *ACM Comput. Surveys*, vol. 45, no. 4, pp. 1–33, Aug. 2013.

[5] E. García-Martín, C. F. Rodrigues, G. Riley, and H. Grahn, ‘‘Estimation of energy consumption in machine learning,’’ *J. Parallel Distrib. Comput.*, vol. 134, pp. 75–88, Dec. 2019.

[6] B. Guidi, K. G. Kapanova, K. Koidl, A. Michienzi, and L. Ricci, ‘‘The contextual ego network P2P overlay for the next generation social networks,’’ *Mobile Netw. Appl.*, vol. 25, no. 3, pp. 1062–1074, Jun. 2020.

[7] L. Mui, M. Mohtashemi, and A. Halberstadt, ‘‘A computational model of trust and reputation,’’ in *Proc. 35th Annu. Hawaii Int. Conf. Syst. Sci.*, Jan. 2002, pp. 2431–2439.

[8] S. Nepal, W. Sherchan, and C. Paris, ‘‘STrust: A trust model for social networks,’’ in *Proc. IEEE 10th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Nov. 2011, pp. 841–846.

[9] D. Olmedilla, O. F. Rana, B. Matthews, and W. Nejdl, ‘‘Security and trust issues in semantic grids,’’ in *Semantic Grid: The Convergence of Technologies*, vol. 5271. Schloss Dagstuhl, Germany: Internationales Begegnungsund Forschungszentrum für Informatik (IBFI), 2006, pp. 1–11.

[10] G. Liu, Y.Wang, and M. Orgun, ‘‘Trust inference in complex trust-oriented social networks,’’ in *Proc. Int. Conf. Comput. Sci. Eng.*, Aug. 2009, pp. 996–1001.