Android Malware Detection Using Supervised Machine Learning

Jai Chiranjeeva Dadi
Amrita Vishwa Vidyapeetam(CSE)
Chennai, India
jaichiranjeeva91@gmail.com

Kousik Parimi
Amrita Vishwa Vidyapeetam(CSE)
Chennai, India
koushik2parimi@gmail.com

Jahnavi Penugonda
Amrita Vishwa Vidyapeetam(CSE)
Chennai, India
jahnavipenugonda892@gmail.com

V.Y.A.M.Karthik Mutyala
Amrita Vishwa Vidyapeetam(CSE)
Chennai, India
karthik.86516@gmail.com

Dr.E Sophiya
Amrita Vishwa Vidyapeetam
Chennai, India
e_sophiya@ch.amrita.edu

***Abstract*—This research paper delves into the training of machine learning models using XGBoost and randomized trees algorithms. These datasets were obtained through both static & dynamic analysis of samples which includes malicious packages and benign packages . We then conducted a comparison of their success rates, with algorithms like forest, decision tree. In our work on the datasets we also compared these algorithms. The performing classification models, utilizing the XGBoost algorithm achieved an accuracy rate of 97% and precision rate of 96% and an F1 score of 0.95 on analysing the dataset. The area under ROC curve is 0.99. Subsequently we exported these performing machine learning models to be utilized in our proposed model for automating the process of dynamic analysis while enabling classification, on new samples.**

I. Introduction

The advent of technology in our on-line world has indeed added various safety concerns especially in our on-line world and generally elsewhere. Interestingly, a particular kind of malware called Android malware has gained recognition, this is essentially due to its rampant use specifically inside the telephone zone. The excessive penetration of malware can be characteristic to the ever increasing quantity of packages available through structures just like the Google Store plus the rising reputation of Android as an opescore machine. Moreover, technologies outside our on-line world have also caused several security problems, each specific to our on-line world and in a broader context. Nonetheless, a particular version of malware called Android malware has gained recognition, predominantly in the phone sector.Researchers are passionately working on locating effective ways to stop malware on Android devices.

They've developed two key detection techniques: Static analysis pursuits to identify harmful behaviors in software program applications without needing to run them, while dynamic analysis appears at conduct that unfolds even as the software is in use. However, each method comes with its personal drawbacks like falling brief in detecting disguised code.

We're introducing a proposed model that cleverly uses static analysis with device gaining knowledge of methods. This method is designed to enhance detection performance. The detection system tackles key demanding situations associated with handling enormous data files in Android and choosing the maximum effective capabilities.

Currently we are working on Permission-primarily based actually malware detection: This method centers on the permissions granted to an software program sooner or later of set up on Android gadgets. The permissions granted dictate an software program software's get proper of access to rights, like having access to contacts or sending statistics over the internet. Some programs request needless permissions. By reading the permission combinations asked through benign and malware programs, it turns into viable to differentiate between benign and malicious software program application. This is achieved through training device learning models using records from regarded benign and malware applications.

II. Literature survey

Fahad et al[1] recommended a permissions-primarily based completely malware detection device that determines the App's maliciousness based totally on the use of suspicious permissions. This tool makes use of a multidegree based totally totally method; First he extracted and choose out the highquality abilities together with permissions, small sizes, and permission charges from a manually collected dataset of 10,000 packages. Further, he employed various device reading models to categorize the Apps into their malicious or benign training. Through notable experimentations, this proposed approach successfully identifies the 5times maximum outstanding capabilities to expect malicious Apps. The proposed method outperformed the triumphing strategies by manner of the use of achieving high accuracies of malware detection i.e., 89.7% with Support Vector Machine, 89.96% with Random Forest,86.25% with Rotation Forest, and 89.52%

with Naïve Bayes models. Moreover, the proposed technique optimized as 70% of the function set in comparison to the current techniques, at the same time as enhancing the assessment metrics which incorporates precision, sensitivity, accuracy, and F-degree. The experimental effects show that the proposed device gives a immoderate stage of symmetry between beside the factor permissions and malware Apps. Further, the proposed device is promising and may offer a low-rate possibility for Android malware detection for malicious or repackaged Apps

Durmus et al[2] proposed a machine learning based malware detection system to distinguish between Android malware and benign applications. In the feature selection step of the proposed malware detection system, it aims to remove redundant features by linear regression-based feature selection method thus reducing feature vector dimension, reducing training time, and classification model can be consumed role in real-time malware detection systems when evaluating academic results At least Using 27 items, the highest score of 0.961 is obtained according to the F-measure

Yildiz et al[3] proposed a method for detecting Android malware the use of feature selection with genetic algorithm (GA). Three different classifier methods with specific characteristic subsets that were selected using GA had been applied for detecting and reading Android malware relatively. A combination of Support Vector Machines and a GA yielded the quality accuracy end result of 98.45% with the sixteen decided on permissions the use of the dataset of 1740 samples consisting of 1119 malwares and 621 benign samples.

Mohamad Arif et al[4] proposed system gaining knowledge of with different units of classifiers become used to assess Android malware detection. The characteristic choice approach in this have a look at was applied to determine which features have been most able to distinguishing malware. A overall of 5000 Drebin malware samples and 5000 Androzoo benign samples were utilised. The performances of the different units of classifiers were then as compared. The results indicated that with a TPR fee of 91.6%, the Random Forest set of rules achieved the best level of accuracy in malware detection.

J Mcdonald et al[5] research investigates effectiveness of 4 special gadget mastering algorithms along side features selected from Android show up record permissions to categorise programs as malicious or benign. Case examine results, on a take a look at set along with 5,243 samples, produce accuracy, don't forget, and precision quotes above 80%. Of the taken into consideration algorithms (Random Forest, Support Vector Machine, Gaussian Naïve Bayes, and K-Means), Random Forest completed the excellent with 82.5% precision and eighty one.Five

III. Methodology

A. Dataset

In this study we utilized a dataset sourced from Mendeley Data, curated by Aravind Mahindru. The dataset comprises 18,850 entries encompassing 175 attributes, collected from sources to detect malware in Android devices. It encompasses both Android application packages and malware infected Android packages providing insights, into the patterns of malicious applications and the permissions they require during runtime.

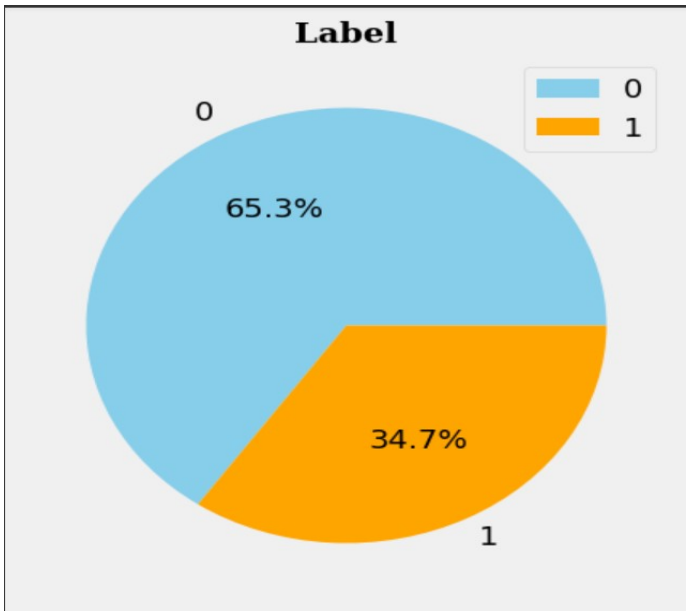

Fig. 1. Pie chart Distribution of Permissions Data

B. Data pre-processing

1) Data Cleaning: There are no missing values in the dataset.

2) Data Integration: The Begnin and Malware data is combined and formed in to a single dataset. While combining the two datasets we have created a new feature called class and assigned 0 for all Begnin instances and 1 for all malware instances.

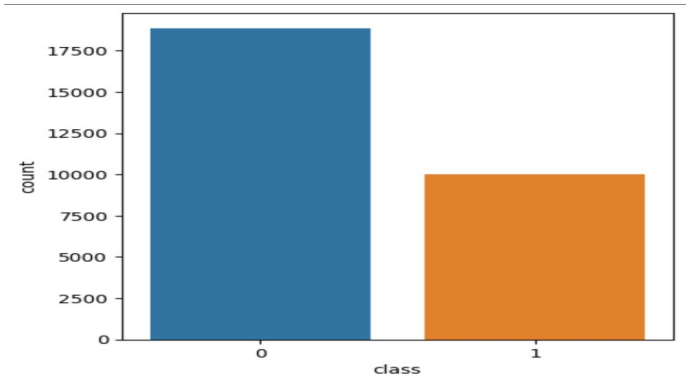

Fig. 2. Class Distribution of Permissions Data

From the above visualization we can infer that 0 represents begnin where as 1 represents malware .

3) Data Transformation: In our data set all features are categorical. Other than package, category and class all features contains 0 and 1 as values. All three features are encoded using label encoder.

C. Model training and evaluation

1) Logistic Regression: Logistic regression, a technique in machine learning, is a way used for classification tasks, not numerical predictions. It estimates the chance of an event going on through analyzing functions in the facts.Logistic regression is preferred for its simplicity and interpretability, making it a useful tool for lots fields despite assumptions approximately linear relationships between capabilities and effects.
The accuracy achieved after using the decision tree is 91% and precision and recall values are 92% and 88% respectively.

2) Support Vector Machine: SVM stand as a strong and flexible method within the realm of system learning. It's a supervised learning algorithm broadly speaking used for category duties, but it's also powerful for regression and outlier detection.
The accuracy acheieved is 87%, and precision, recall values are of 90%, 82% respectively when the Support Vector Machine model was used.

3) Decision Tree: Decision tree is used for solving category and regression issues.Decision tree can manage both numerical and categorical data and cope with missing values with out much preprocessing.The algorithm builds tree-like a model, where each node in the tree represents a decision, and each branch represents the outcome of that decision. The leaf nodes of the tree represent the final predictions made by the model.

	precision	recall	f1-score
0	0.97	0.96	0.97
1	0.93	0.94	0.94
accuracy			0.96
macro avg	0.95	0.95	0.95
weighted avg	0.96	0.96	0.96

Fig. 3. confusion matrix for Decision Tree

The accuracy achieved after using the decision tree is 96% and precision and recall values are 95% and 95% respectively.

4) Random Forest: The Random Forest represents an ensemble learning technique. During its training phase, this algorithm builds numerous decision trees and then provides the class that either appears most frequently among the classes or the average prediction derived from the individual trees.
we got an accuracy of 96%, a precision of 95%, and recall of 94%, when the Random Forest model was used.

	precision	recall	f1-score
0	0.95	0.99	0.97
1	0.97	0.90	0.94
accuracy			0.96
macro avg	0.96	0.94	0.95
weighted avg	0.96	0.96	0.96

Fig. 4. confusion matrix for Random Forest

5) XGBoost Classifier: XGBoost classifier is an efficient machine learning model. It is widely used in supervised learning tasks.XGBoost classifier is known for its speed and efficiency and its ability to handle false positives and false negatives.

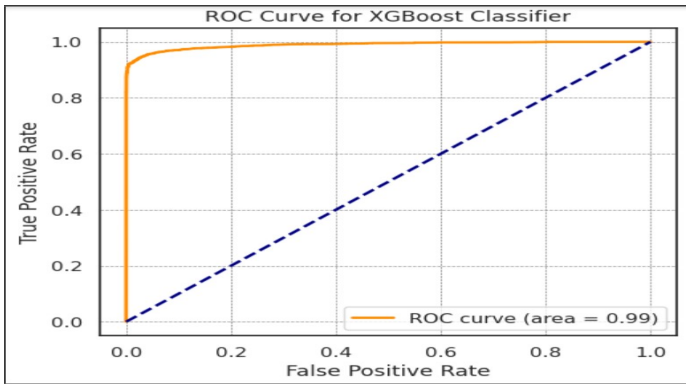

Fig. 5. ROC curve for XGBoost classifier

	precision	recall	f1-score	support
0	0.96	0.99	0.98	6209
1	0.98	0.92	0.95	3312
accuracy			0.97	9521
macro avg	0.97	0.96	0.96	9521
weighted avg	0.97	0.97	0.97	9521

Fig. 6. confusion matrix for XGBoost classifier

The XGBoost Classifier achieved an accuracy of 97%, a precision of 97%, and a recall of 96%.

IV. Results

Out of all the models XGboost classifier performed well with an accuracy 0.97. And other evaluation metrics like area under Roc curve , precision, recall,f1 score are also pretty good.

V. Future Scope & Conclusion

So, we propose a malware detection system using Xgboost classifier. We can increase the accuracy using hyper parameter tuning.We can also add feature like how frequently it taking

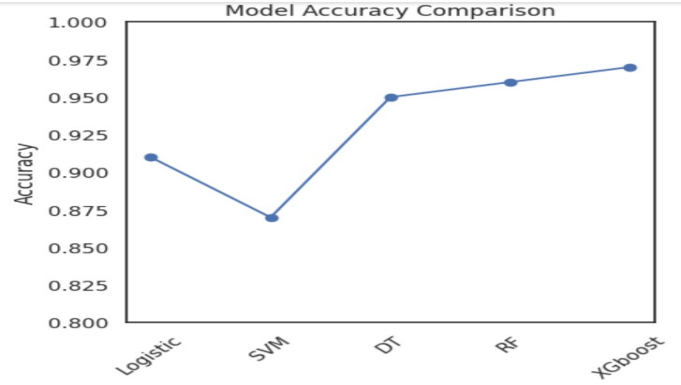

Fig. 7. Line plot showing all models accuracies

permissions from user and running in the background. We can also try to predict the andriod malware using data of app behaviour.

REFERENCES

[1] Akbar, F.; Hussain, M.;Mumtaz, R.; Riaz, Q.; Wahab, A.W.A.;Jung, K.-H. Permissions-Based Detection of Android Malware Using Machine Learning. Symmetry 2022,14, 718. https://doi.org/10.3390/sym14040718

[2] sahin, Durmus Kural, Oguz Akleylek, Sedat Kilic, Erdal. (2021). A novel permission-based Android malware detection system using feature selection based on linear regression. Neural Computing and Applications. 35. 1-16. 10.1007/s00521-021-05875-1.

[3] Yildiz, Oktay Doğru, İbrahim. (2019). Permission-based Android Malware Detection System Using Feature Selection with Genetic Algorithm. International Journal of Software Engineering and Knowledge Engineering. 29. 245-262. 10.1142/S0218194019500116.

[4] Mohamad Arif J, Ab Razak MF, Awang S, Tuan Mat SR, Ismail NSN, et al. (2021) A static analysis approach for Android permission-based malware detection systems.

[5] Machine Learning-Based Android Malware Detection Using Manifest Permissions Proceedings of 54th Hawaii International Conference on System Sciences (HICSS-54), Kauai, Hawaii, January 5-8 2021. Nathan Herron, J. Todd McDonald, William B. Glisson, and Ryan Benton.

[6] K. Liu, S. Xu, G. Xu, M. Zhang, D. Sun and H. Liu, "A Review of Android Malware Detection Approaches Based on Machine Learning," in IEEE Access, vol. 8, pp. 124579-124607, 2020, doi: 10.1109/ACCESS.2020.3006143.

[7] A. Droos, A. Al-Mahadeen, T. Al-Harasis, R. Al-Attar and M. Ababneh, "Android Malware Detection Using Machine Learning," 2022 13th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2022, pp. 36-41, doi: 10.1109/ICICS55353.2022.9811130.

[8] R. Agrawal, V. Shah, S. Chavan, G. Gourshete and N. Shaikh, "Android Malware Detection Using Machine Learning," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 2020, pp. 1-4, doi: 10.1109/ic-ETITE47903.2020.491.

[9] M. P. Singh and H. K. Khan, "Malware Detection in Android Applications Using Machine Learning," 2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS), Bangalore, India, 2023, pp. 105-110, doi: 10.1109/ICAECIS58353.2023.10170311.

[10] S. Sabhadiya, J. Barad and J. Gheewala, "Android Malware Detection using Deep Learning," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 1254-1260, doi: 10.1109/ICOEI.2019.8862633.

[11] J. Sahs and L. Khan, "A Machine Learning Approach to Android Malware Detection," 2012 European Intelligence and Security Informatics Conference, Odense, Denmark, 2012, pp. 141-147, doi: 10.1109/EISIC.2012.34.

[12] R. Kuchipudi, M. Uddin, T. S. Murthy, T. K. Mirrudoddi, M. Ahmed and R. B. P, "Android Malware Detection using Ensemble Learning," 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2023, pp. 297-302, doi: 10.1109/ICSCSS57650.2023.10169578.

[13] A. Fatima, R. Maurya, M. K. Dutta, R. Burget and J. Masek, "Android Malware Detection Using Genetic Algorithm based Optimized Feature Selection and Machine Learning," 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), Budapest, Hungary, 2019, pp. 220-223, doi: 10.1109/TSP.2019.8769039.

[14] J. D. Koli, "RanDroid: Android malware detection using random machine learning classifiers," 2018 Technologies for Smart-City Energy Security and Power (ICSESP), Bhubaneswar, India, 2018, pp. 1-6, doi: 10.1109/ICSESP.2018.8376705.

[15] S. Y. Yerima, S. Sezer and I. Muttik, "Android Malware Detection Using Parallel Machine Learning Classifiers," 2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies, Oxford, UK, 2014, pp. 37-42, doi: 10.1109/NGMAST.2014.23.