ANDROID MALWARE DETECTION USING GENETIC ALGORITHM BASED OPTIMISED FEATURE SELECTION AND MACHINE LEARNING

[1]K. Jaya Krishna, [2]Anna Adi Kalyani,

[1]Associate Professor, Department of Master of Computer Applications,
QIS College of Engineering & Technology, Ongole, Andhra Pradesh, India
[2]PG Scholar, Department of Master of Computer Applications,
QIS College of Engineering & Technology, Ongole, Andhra Pradesh, India

ABSTRACT

Malware is one of the major issues regarding the operating system or in the software world. The android system is also going through the same problems. We have seen other Signature-based malware detection techniques were used to detect malware. But the techniques were not able to detect unknown malware. Despite numerous detection and analysis techniques are there, the detection accuracy of new malware is still a crucial issue. In this paper, we study and highlight the existing detection and analysis methods used for the android malicious code. Along with studying, we propose Machine learning algorithms that will be used to analyze such malware and also we will be doing semantic analysis. We will be having a data set of permissions for malicious applications. Which will be compared with the permissions extracted from the application which we want to analyze. In the end, the user will be able to see how much malicious permission is there in the application and also we analyze the application through comments.

I.INTRODUCTION

Malware is one of the major issues regarding the operating system or in the software world. The android system is also going through the same problems. We have seen other Signature-based malware detection techniques were used to detect malware. But the techniques were not able to detect unknown malware. Despite numerous detection and analysis techniques are there, the detection accuracy of new malware is still a crucial issue. In this paper, we study and highlight the existing detection and analysis methods used for the android malicious code. Along with studying, we propose Machine learning algorithms that will be used to analyze such malware and also we will be doing semantic analysis. We will be having a data set of permissions for malicious applications. Which will be compared with the permissions extracted from the application which we want to analyze. In the end, the user will be able to see how much malicious permission is there in the application and also we analyze the application through comments. Android platform due to open source characteristic and Google backing has the largest global market share. Being the world's most popular operating system, it has drawn the attention of cyber criminals operating particularly through wide distribution of malicious applications. This paper proposes an effectual machine-learning based approach for Android Malware Detection making use of evolutionary Genetic algorithm for discriminatory feature selection. Selected features from Genetic algorithm are used to train machine learning classifiers and

their capability in identification of Malware before and after feature selection is compared. The experimentation results validate that Genetic algorithm gives most optimized feature subset helping in reduction of feature dimension to less than half of the original feature-set. Classification accuracy of more than 94% is maintained post feature selection for the machine learning based classifiers, while working on much reduced feature dimension, thereby, having a positive impact on computational complexity of learning classifiers.

II.EXISITNG SYSTEM

In the existing system, the application permissions are extracted to detect the malware and executed through the command prompt. A proper GUI was not provided to execute the tasks. All the commands were run through the command prompt. It was difficult for the non-technical user to use the system. And also Semantic analysis was not implemented.

III.PROPOSED SYSTEM

In the proposed system, we are doing the permission-based analysis and also the semantic analysis. The permission-based analysis is been done on the web-based UI while the existing systems were just doing it all on the local machine in the command prompt. In our system, we have implemented an admin panel as well as a user panel. In the admin panel admin have the access to upload the apk files and its details along with its categorization and also the admin can upload the comment that can be used for semantic analysis.

In the user-panel the user can see the select the category of the application and can see its details like pricing description name. User can see the malicious percentage of the application. And the processed output of the semantic analysis will be displayed to the user in the form of graph and the user will get a proper review of the application.

1. DATA COLLECTION MODULE:

- Gathers Android application data from various sources (app stores, repositories, etc.) for analysis.
- Collects both malware and benign app samples for training and testing.

2. FEATURE EXTRACTION MODULE:

- Extracts relevant features from Android apps (permissions, API calls, intents, etc.) to represent them for analysis.
- Transforms raw data into a format suitable for machine learning-based analysis.

3. ALGORITHM MODULE

- Implements the genetic algorithm framework for feature selection or optimization. Applies evolutionary techniques to refine the feature set and improve classification accuracy.

4. MACHINE LEARNING MODEL MODULE

- Develops machine learning models (e.g., SVM, Random Forest, Neural Networks) for classification.
- Trains models using extracted features to differentiate between malware and benign apps.

5. DETECTION MODULE:

- Implements a module capable of real-time Android malware detection on devices.
- Deploys a lightweight version of the detection model for on-device scanning.

6.VISUALIZATION MODULE:

- Generates reports showcasing analysis results, model performance, and detected malware instances.
- Provides visualizations for easy comprehension of data and analysis outcomes.

IV.ALGORITHMS

Logistic regression Classifiers

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name logistic regression is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name multinomial logistic regression is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

Logistic regression competes with discriminant analysis as a method for analyzing categorical-response variables. Many statisticians feel that logistic regression is more versatile and better suited for modeling most situations than is discriminant analysis. This is because logistic regression does not assume that the independent variables are normally distributed, as discriminant analysis does.

This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots. It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables. It provides confidence intervals on predicted values and provides ROC curves to help determine the best cutoff point for classification. It allows you to validate your results by automatically classifying rows that are not used during the analysis.

Algorithm 2: PSEUDO code for logistic regression algorithm

```
Step1: Function grad (predictor_attributes, target_attribute, weights)
            {
              Calculate gradient_descent;
              Return weights + learning_rate * gradient_descent;
            }
Step2: Normalize the dataset;
Step3: Repeat
    {
      Weights = grad (params);
      Update weights;
    } until convergence
Step4: z = dot product of predictor variables and updated weights;
Step5: prediction_limit = sigmoid function (z);
Step6: Predict the target class
```

Support Vector Machine (SVM):

Support Vector Machine (SVM) is a powerful supervised learning algorithm used for classification and regression tasks. SVM operates by finding the optimal hyperplane that best separates classes in the feature space, maximizing the margin

between classes. For linearly separable data, SVM aims to find a hyperplane that maximizes the distance between the closest data points of different classes. The decision function of SVM for classification can be formulated as:

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N_{SV}} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b\right)$$

Here, x\mathbf{x}x represents the input vector to be classified, NSVN_{\text{SV}}NSV denotes the number of support vectors, xi\mathbf{x}_ixi are the support vectors, yiy_iyi are their corresponding class labels, αi\alpha_iαi are Lagrange multipliers, K(xi,x)K(\mathbf{x}_i, \mathbf{x})K(xi,x) is the kernel function that maps the input space into a higher-dimensional feature space, and bbb is the bias term. SVMs are effective in handling high-dimensional data and are particularly useful when the number of features exceeds the number of samples. They are widely used in applications such as text categorization, image classification, and bioinformatics due to their ability to handle complex data distributions and nonlinear decision boundaries.

Algorithm 1: SVM

1. Set $Input = (x_i, y_i)$, where $i = 1, 2, \cdots, N, x_i = R^n$ and $y_i = \{+1, -1\}$.
2. Assign $f(X) = \omega^T x_i + b = \sum_{i=1}^{N} \omega^T x_i + b = 0$
3. Minimize the QP problem as, $min\ \varphi(\omega, \xi) = \frac{1}{2}\|\omega\|^2 + C.(\sum_{i=1}^{N} \xi_i)$.
4. Calculate the dual Lagrangian multipliers as $min\ L_P = \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{N} x_i y_i(\omega x_i + b) + \sum_{i=1}^{N} x_i$.
5. Calculate the dual quadratic optimization (QP) problem as $max\ L_D = \sum_{i=1}^{N} x_i - \frac{1}{2}\sum_{i,j=1}^{N} x_i x_j\ y_i\ y_j(x_i, x_j)$.
6. Solve dual optimization problem as $\sum_{i=1}^{N} y_i x_i = 0$.
7. Output the classifier as $f(X) = sgn(\sum_{i=1}^{N} x_i y_i(x \cdot x_i) + b$.

Neural Network (NN):

Neural Networks (NNs) are computational models inspired by the structure and functioning of the human brain's neural networks. NNs consist of interconnected nodes (neurons) organized in layers: input layer, hidden layers (which can be multiple), and output layer. Each neuron processes its input through weighted connections, applies an activation function, and passes the output to the next layer. The forward propagation process in a neural network can be described mathematically as:

$$a_j^l = \phi\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right)$$

Training a neural network involves forward propagation to compute predictions and backward propagation (backpropagation) to update weights using gradient descent or its variants. NNs are highly flexible and capable of learning complex patterns and relationships in data, making them suitable for tasks such as image recognition, natural language processing, and reinforcement learning. However, NNs require significant computational resources and careful parameter tuning to achieve optimal performance

V.RESULT

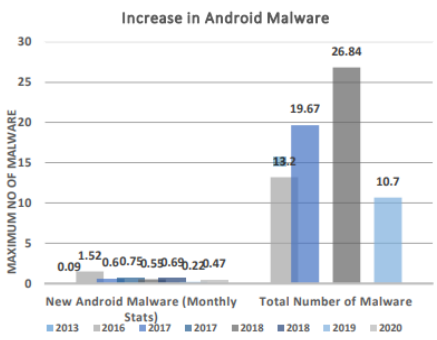

Fig1:Third party well known dangerous apps increase

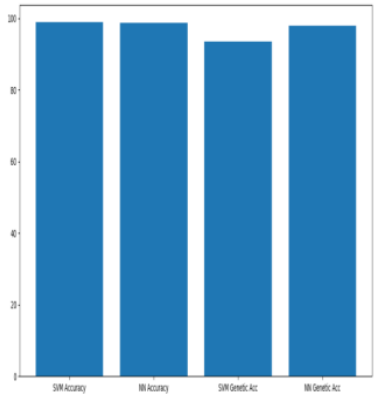

Fig2: Accuracy Graph

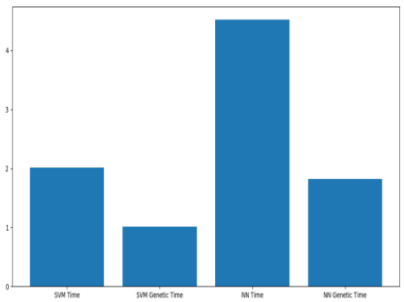

Fig3: Execution Time Graph

VI CONCLUSION

In our work, we propose a system for permission analysis and semantic analysis. Our system is also used to detect malware permissions based on an application by comparing it with a dataset. This proposed system can be applied in the fields of the security system and also for the n users like a malware detection software. However, there are limitations in our system. The permissions which we are defining are as per our but it can differ from users to users. The permissions which the user likes that it is not a malware-based can be malware for any other user. Future works will contain the improvement of that.

VII.REFERENCES

- D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon and K. Rieck, "Drebin: Effective and Explainable Detection of Android Malware in Your Pocket", *Proceedings 2014 Network and Distributed System Security Symposium*, 2014.
- N. Milosevic, A. Dehghantanha and K. K. R. Choo, "Machine learning aided Android malware classification", *Comput. Electr. Eng.*, vol. 61, pp. 266-274, 2017.
- J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-An and H. Ye, "Significant Permission Identification for Machine-Learning-Based Android Malware Detection", *IEEE Trans. Ind. Informatics*, vol. 14, no. 7, pp. 3216-3225, 2018.
- A. Saracino, D. Sgandurra, G. Dini and F. Martinelli, "MADAM: Effective and Efficient Behavior-based Android Malware Detection and Prevention", *IEEE Trans. Dependable Secur. Comput.*, vol. 15, no. 1, pp. 83-97, 2018.
- S. Arshad, M. A. Shah, A. Wahid, A. Mehmood, H. Song and H. Yu, "SAMADroid: A Novel 3-Level Hybrid Malware Detection Model for Android Operating System", *IEEE Access*, vol. 6, pp. 4321-4339, 2018.
- T. Kim, B. Kang, M. Rho, S. Sezer and E. G. Im, "A Multimodal Deep Learning Method for Android Malware Detection using Various Features", vol. 6013, no. c, 2018.
- A. Martin, F. Fuentes-Hurtado, V. Naranjo and D. Camacho, "Evolving Deep Neural Networks architectures for Android malware classification", *2017 IEEE Congr. Evol. Comput.*

CEC 2017-Proc., pp. 1659-1666, 2017.

- X. Su, D. Zhang, W. Li and K. Zhao, "A Deep Learning Approach to Android Malware Feature Learning and Detection", *2016 IEEE Trust*, pp. 244-251, 2016.
- K. Zhao, D. Zhang, X. Su and W. Li, "Fest: A Feature Extraction and Selection Tool for Android Malware Detection", *2015 IEEE Symp. Comput. Commun*, pp. 714-720, 4893.
- A. Feizollah, N. B. Anuar, R. Salleh and A. W. A. Wahab, "A review on feature selection in mobile malware detection", *Digit. Investig.*, vol. 13, pp. 22-37, 2015
- A. Firdaus, N. B. Anuar, A. Karim, M.Faizal and A. Razak, Discovering optimal features using static analysis and a genetic search based method for Android malware vol. 19, no. 6, pp. 712-736, 2018.

A. V. Phan, M. Le Nguyen and L. T. Bui, "Feature weighting and SVM parameters optimization based on genetic algorithms for classification problems", *Appl. Intell.*, vol. 46, no. 2, pp. 455-469, 2017

Authors

[1] Mr. K. Jaya Krishna, currently working as an Associate Professor in the Department of Master of Computer Applications, QIS College of Engineering and Technology, Ongole, Andhra Pradesh. He did his MCA from Anna University, Chennai, M.Tech (CSE) from JNTUK, Kakinada. He published more than 10 research papers in reputed peer reviewed Scopus indexed journals. He also attended and presented research papers in different national and international journals and the proceedings were indexed IEEE. His area of interest is Machine Learning, Artificial intelligence, Cloud Computing and Programming Languages.

[2] Ms. Anna Adi Kalyani, currently pursuing Master of Computer Applications at QIS College of engineering and Technology (Autonomous), Ongole, Andhra Pradesh. She Completed B.Sc. in Computer Science from Sri Sai Degree College, Vinukonda, Andhra Pradesh. Her areas of interest are Machine learning&Python.