Comparative Analysis on Usage Of Machine learning And Deep Learning In Android Malware detection.

Regidi AjayKumar , Ms N KrishnaVeni (Assistant Professor ,GMRIT,Rajam)

**Abstract:**

The Android operating system has been the most popular for smartphones and tablets . This popularity has led to a rapid raise of Android malware in recent years.The android can get safety by detecting malware. There are various deep learning (DL) and machine learning (ML) methods used to detect malware in the Android operating system. As the use of Android applications has grown, so has the threat of cyberattacks, particularly through malware. Traditional detection methods, such as signature-based techniques, are proving inadequate against the rapidly evolving nature of malware, especially in real-time. DL models like convolutional neural networks , gated recurrent units (GRUs), and long short-term memory (LSTM) and machine learning models like NB,SVM,RandomForest,DT are increasingly being used for malware detection. These models show promise in real-time detection, but challenges persist in monitoring malware behavior and maintaining updated datasets. Emphasizes the importance of training users and enhancing data-sharing mechanisms to strengthen Android security. The deep learning models are increasingly becoming an effective technique for malicious software detection in Android applications in realtime. Improving the malware detection capabilities may help for improvement of future malware detection in android.Android malware detection plays a major role in detecting and solving the malware issues in android.

**Keywords:**

*Deep Learning, Malware detection, CNN , GRU , LSTM , DNN.*

**Introduction:**

The Android operating system has emerged as the dominant platform for smartphones and tablets, it has also become a prime target for cyberattacks, particularly through malware. The open nature and widespread adoption of Android applications have significantly increased the exposure to various security threats.

Traditional malware detection methods, such as signature-based approaches etc. The need for more sophisticated techniques to ensure the safety of Android devices is now more crucial than ever. In response to these challenges, deep learning (DL) and machine learning (ML) methods are being increasingly applied to Android malware detection. These models, such as convolutional neural networks, deep neural networks, long short-term memory and gated recurrent units, have shown significant promise in detecting malware in real-time scenarios. There are various deep learning (DL) and machine learning (ML) methods used to detect malware in the Android operating system. As the use of Android applications has grown, so has the threat of cyberattacks, particularly through malware. Unlike traditional methods, DL models can dynamically adapt to new malware behaviors, making them more effective in combating the growing and ever-changing threat landscape. Improving detection mechanisms through the integration of advanced DL models could have the way for more secure Android environments, reducing the risk of malware in android.

**Literature Review:**

This study explores Android malware detection through a deep learning-based model, using a dataset of Android apps. It highlights how gated recurrent units(GRU) are applied to identify malicious software .[1]

This evaluates the performance of various machine learning classifiers, including Support Vector Machines (SVM), for Android malware detection using system-level features. By analyzing characteristics such as system calls and resource usage, the study aims to distinguish between benign and malicious applications effectively. The highlight is SVM's robustness in high-dimensional spaces, contributing valuable insights for enhancing mobile security.[2]

Advancements in Android malware detection include the introduction of a Random Forest machine learning model that focuses on system permissions and API calls as key features. The study evaluates various classifiers, analyzing their strengths and weaknesses in identifying malicious applications. This approach leverages the unique characteristics of Android apps to improve detection accuracy and robustness.[3]

This work presents a dynamic permissions-based Android malware detection approach utilizing a Simple Logistic model for machine learning. It emphasizes the importance of analyzing permissions requested by applications during runtime to facilitate real-time malware identification.[4]

This innovative approach focuses on utilizing the unique visual representations of application behavior, potentially improving detection accuracy. The findings highlight the promise of image-based analysis in enhancing mobile security and developing robust malware detection systems for Android platforms.[5]

These results contribute to the development of robust detection systems since they enhance mobile security by using behavioural characteristics of apps due to their permissions and API interactions.[6]

This paper presents techniques for Android malware detection with a particular emphasis on the machine learning Support Vector Machine (SVM) model. It explores three detection methodologies: static, dynamic, and hybrid approaches, thoroughly analyzing their effectiveness and limitations in addressing malware threats.[7]

MalDozer is a deep learning-based framework specifically designed for Android malware detection. It employs static analysis to characterize malware by extracting relevant features from applications, achieving high detection accuracy in identifying malicious software.[8]

This paper also examines deep learning methods, including the MalDozer framework, for Android malware detection. It analyzes static, dynamic, and hybrid detection techniques, discussing their respective strengths and weaknesses while offering recommendations for future research directions in the field.[9]

Long Short-Term Memory (LSTM) model for detecting Android malware. It evaluates the model's performance based on system-level features extracted from Android applications, highlighting its effectiveness in identifying malicious behavior.[10]

This compares various machine learning models, such as Random Forest and Naive Bayes, for Android malware detection. It analyzes their strengths and limitations when applied to static features and benchmarks the classifiers using standard performance metrics.[11]

The model focuses on the analysis of real-time data streams generated by Android applications, allowing for the immediate identification and response to malicious activities. By leveraging continuous monitoring and dynamic analysis, the system aims to detect malware as it executes, rather than on static analysis prior to installation.[12]

This investigates deep learning techniques for Android malware detection, emphasizing the importance of feature extraction in capturing complex malware patterns. The study highlights the effectiveness of deep learning in enhancing detection capabilities.[13]

This concentrating on feature extraction methods and their impact on classifier performance. It compares different models to assess their strengths and weaknesses in detecting malware.[14]

The paper discusses the architecture of the DL-Droid system, detailing how it captures and processes various runtime features, such as API calls, system logs, and application interactions. This approach enables the system to identify suspicious patterns and behaviors indicative of malware activity, providing a more comprehensive detection mechanism [15]

**Methodology:**

The methodology mainly focuses on detecting Android malware using a combination of machine learning and deep learning algorithms and these algorithms that evaluate by taking key features that are based on API calls and permissions.The algorithms are

**Machine learning algorithms:**

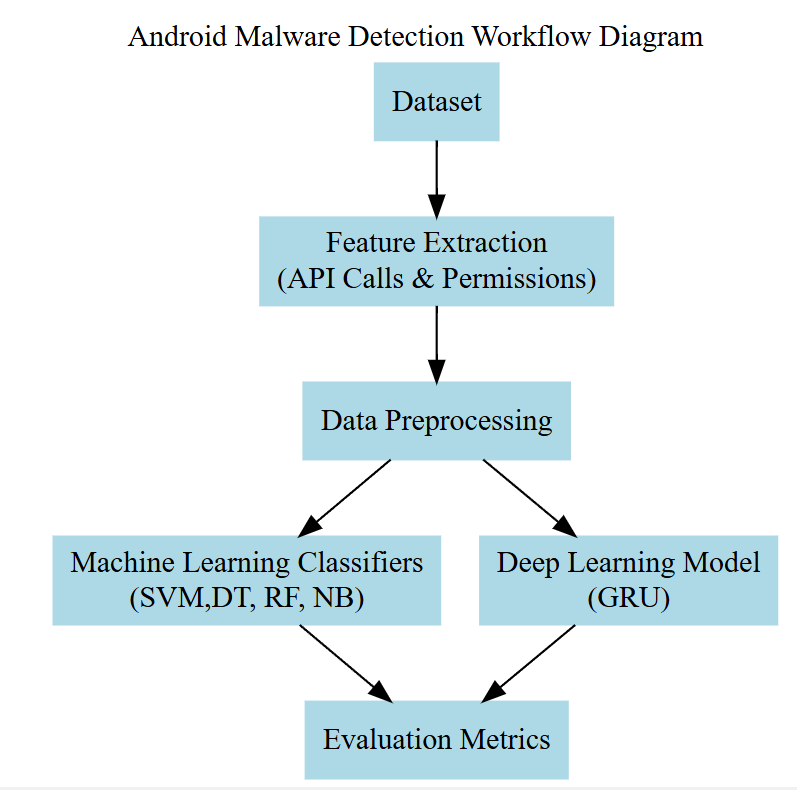
* Support vector machine
* Random Forest
* Naive Bayes
* Decision Tree

**Deep learning algorithms:**

GRU(gated recurrent unit)

Here detecting can be done by taking the CICAndMal2017 dataset, which consists of benign app’s samples and malware samples divided into categories like Adware,Ransomware,Adsware, and SMSware.

* **Naive Bayes (NB):**
* Naive Bayes is a probabilistic classifier based on Bayes Theorem, which assumes that features are independent given the class label.
* Naive bayes computes the the probability of an application being malicious or benign with the help of its features, such as API calls and permissions. It learns the probability of occurrence of each feature within a class during the training phase with the help of a dataset consisting of malware and benign applications.
* **Decision Tree (DT):**
* A Decision Tree (DT) classifier detects malware in Android applications by first training on a labeled dataset that includes both benign and malware samples.It constructs a tree structure by selecting features, such as API calls and permissions.
* In classification, the Decision Tree traverses the tree from the root to a leaf node in order to determine the features of a application. At each node, decisions are made based on the feature values. The application is classified as either malware or benign depending on the majority class of the samples that reach the leaf node. This enables decision-making based on the characteristics of the application.
* **Support Vector Machine:**
* The Support Vector Machine (SVM) classifier detects malware in Android applications by first training on a dataset containing both benign and malware samples,where features such as API calls and permissions are extracted.
* SVM works by finding the optimal hyperplane that separates the two classes like malware and benign in a high-dimensional feature space. During the training phase, it identifies support vectors, which are the data points closest to the hyperplane. When classifying a new application, SVM determines which side of the hyperplane the application’s feature vector falls on, thus classifying it as either malware or benign.This method is effective in handling high-dimensional data.
* **Random Forest:**
* The Random Forest (RF) classifier detects malware in the android applications based on the construction of thousands of decision trees during training. In this approach, every tree is constructed using a random subset of the available training data and features, hence encouraging uniqeness and minimizing overfitting.
* The classification is determined by, each tree in the forest makes a prediction about whether the application is malware or benign, and the final classification is determined by majority voting among all the trees.
* **Gated Recurrent Unit:**
* First, the GRU model collects important information from Android applications, focusing on API calls and permissions.
* During training,the GRU is trained over dataset that includes both benign and malicious apps. During this phase, the model learns to recognize patterns by adjusting its internal settings to reduce errors in its predictions. After training, when a new app is analyzed, the GRU uses what it learned to predict whether the app is benign or malicious. It produces a score that reflects the prediction of the app being malware.
* The final output layer uses a sigmoid activation function to provide a probability score for classification, achieving a high accuracy.



**Conclusion:**

The study underscores the growing importance of machine learning (ML) and deep learning (DL) techniques in enhancing Android malware detection. The integration of advanced DL models like GRU and LSTM demonstrates getting improvements in identifying complex malware behaviors . Research on classifiers such as Random Forest and SVM highlights their effectiveness in feature extraction and malware classification, proving valuable in real-time detection. Innovative approaches, like MalDozer and image-based analysis, further strengthen malware detection capabilities. This also highlights the potential for hybrid detection methods that combine static and dynamic features to provide robust, real-time malware detection. By leveraging both types of data, these hybrid models can enhance detection accuracy and resilience against sophisticated malware that may evade single-method approaches.

**References:**

[1] Elayan, O. N., & Mustafa, A. M. (2021). Android malware detection using deep learning. Procedia Computer Science, 184, 847-852.

[2] Ham, H. S., & Choi, M. J. (2013, October). Analysis of android malware detection performance using machine learning classifiers. In 2013 international conference on ICT Convergence (ICTC) (pp. 490-495). Ieee.

[3] B. TAHTACI and B. CANBAY, "Android Malware Detection Using Machine Learning," 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), Istanbul, Turkey, 2020, pp. 1-6, doi: 10.1109/ASYU50717.2020.9259834.

[4] Mahindru, A., & Singh, P. (2017, February). Dynamic permissions based android malware detection using machine learning techniques. In Proceedings of the 10th innovations in software engineering conference (pp. 202-210).

[5] Darus, F. M., Salleh, N. A. A., & Ariffin, A. F. M. (2018, November). Android malware detection using machine learning on image patterns. In 2018 Cyber Resilience Conference (CRC) (pp. 1-2). IEEE.

[6] Peiravian, N., & Zhu, X. (2013, November). Machine learning for android malware detection using permission and api calls. In 2013 IEEE 25th international conference on tools with artificial intelligence (pp. 300-305). IEEE.

[7] Alqahtani, E. J., Zagrouba, R., & Almuhaideb, A. (2019, June). A survey on android malware detection techniques using machine learning algorithms. In 2019 Sixth International Conference on Software Defined Systems (SDS) (pp. 110-117). IEEE.

[8] Karbab, E. B., Debbabi, M., Derhab, A., & Mouheb, D. (2018). MalDozer: Automatic framework for android malware detection using deep learning. Digital investigation, 24, S48-S59.

[9] Liu, K., Xu, S., Xu, G., Zhang, M., Sun, D., & Liu, H. (2020). A review of android malware detection approaches based on machine learning. IEEE access, 8, 124579-124607.

[10] Agrawal, R., Shah, V., Chavan, S., Gourshete, G., & Shaikh, N. (2020, February). Android malware detection using machine learning. In 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE) (pp. 1-4). IEEE.

[11] Rana, M. S., Gudla, C., & Sung, A. H. (2018, December). Evaluating machine learning models for Android malware detection: A comparison study. In Proceedings of the 2018 VII International Conference on Network, Communication and Computing (pp. 17-21).

[12] Lê, N. C., Nguyen, T. M., Truong, T., Nguyen, N. D., & Ngô, T. (2020, October). A Machine Learning Approach for Real Time Android Malware Detection. In 2020 RIVF International Conference on Computing and Communication Technologies (RIVF) (pp. 1-6). IEEE

[13] Zhang, N., Tan, Y. A., Yang, C., & Li, Y. (2021). Deep learning feature exploration for android malware detection. Applied Soft Computing, 102, 107069.

[14] Hahn, S., Protsenko, M., & Müller, T. (2016). Comparative evaluation of machine learning-based malware detection on android.

[15] Alzaylaee, M. K., Yerima, S. Y., & Sezer, S. (2020). DL-Droid: Deep learning based android malware detection using real devices. Computers & Security, 89, 101663.