Detection of Malware using Machine Learning in Android Devices/Applications

1. Goutham Lingampally (lgoutham7@gmail.com) Author
2. Shiva Krishna Gujju (shivagujju@gmail.com) Co- Author
3. B.Shobini

Abstract

Spreading malware through Android devices and applications became an important strategy of cyber attackers. Therefore, malware detection in Android applications has become an important area of research. In this context, it is important to answer the question that reads "how can we develop a model based on Machine Learning (ML) to detect malware in Android devices/applications?" When malware is detected in real time from Android mobile applications, it can relive the users of Android phones from the risk of malware. It will also help stakeholders of Android devices to be safe from malicious software. The proposed system extracts feature from. APK files and training is given for supervised learning. Different ML models like Multinomial Naïve Bayes, Random Forest and SVM are used as prediction models. With these ML techniques a framework is realized to have provision for protection of malware in Android devices or applications. The proposed solution continues giving support with increased quality. The rationale behind this is that as the applications are protected and malware is detected, the training data gets increased. With increased training data, it will become much more accurate as time goes on. With some changes, it can be made to detect Android applications live when it is associated with a competing device.

Keywords – Malware detection, feature extraction, machine learning, SVM, Random Forest, Multinomial Naïve Bayes

1. INTRODUCTION

Malware detection research has been around for many years. However, in the context of the latest technology known as Internet of Things (IoT), the threat of malware became more apparent. When there are number of IoT applications being used in the real word, each IoT application is made up of many devices and sensor networks. The devices include smart phones as well [11], [12], [14]. As the Android operating system (OS) is used by many smart phones, in this research Android mobiles and applications are considered for malware detection. Machine Learning (ML) based approach is followed for the detection of malware. As mobile devices are increasingly used in the modern applications, it is essential to know the reasons for malware spread and have a methodology for detection of malware.

In this project, malware detection framework is proposed using ML techniques such as SVM, RF and DT. By using the concept of extraction of features from. APK file, the proposed methodology works based on the features learned. The ML models are used to predict the class labels. Multinomial Naïve Bayes

classifier, Random Forest Classifier and SVM classifier are effectively used as underlying models with feature selection in order to improve classification accuracy. The remainder of this paper is structured as follows. Section 2 reviews literature on malware detection methods existing. Section 3 presents the proposed methodology. Section 4 covers the experimental results and Section 5 concludes the paper and gives future scope of the research.

2. RELATED WORK

The use of mobile devices is increasing every year and with that there has been increased number of attacks or malware intrusions on Android mobile applications. Anwar *et al*. [1] proposed focused on mobile botnets. Mobile botnet is one of the threats to mobile devices. Malicious applications associated with Android mobiles target mobile botnets in order to spread malware. A static method is proposed in order to detect botnets in Android applications. This method is made up of different techniques such as broadcast receivers, permissions and MD5. The features of Android applications are extracted and subjected to machine learning. Especially supervised learning approach is followed in order to classify the applications as malware affected or not. They used a dataset known as UNB ISCX. The dataset has around 14 kinds of malwares. The proposed method extracts feature from the mobile applications and then use the features for training a classifier. Then the trained classifier is further used to detect botnets association with applications. Malatras *et al*. [5] investigated on different mobile botnets and the challenges thrown by them. A mobile botnet contains many components such as bot master, server, bots in the form of servants and clients and communication channels. The taxonomy includes mobile botnets, target, attacks and detection measures. Sensors are used as side channel in order to detect botnets and take corrective measures.

There are many deep learning based solutions existed on the malware detection in Android applications. Watcher and Yu [2] made an extensive review of literature on the deep learning based methods. The methods include supervised learning methods with regression and classification, unsupervised learning methods like clustering, dimensionality reduction and density estimation and reinforcement learning methods like policy search, value function, TensorFlow, DeepLearning4J, Theano, Torch and so on. There are different applications of deep learning including malware detection.

Dynamic methods are the methods that change at runtime based on the situation. It is on the contrary to static methods. Hasan *et al*. [3] proposed a methodology towards cyber security. It is known for working good against StuxMob which is a situational-aware malware that targets Android mobile applications. The StuxMob works differently when compared with the existing applications. It has its own threat model and it makes use of payloads based on targets unlike traditional methods that make use of command and control based botnet. The StuxMob makes use of physical activities of mobile users and then plans attacks based on the runtime situations. The targeted attacks include spying, hacktivist, ransomware, advertising, and attacks on health. The StuxMob makes use of sensor data, performs action identification, maintain an action log, send trigger signal and classification is made finally from action log. Han *et al*. [4] proposed a big data and cloud based model for malware detection. It is one of the smart approaches that are available.

Eustance *et al*. [7] focused on security and privacy of mobile devices. They considered self-awareness, human factors and intervention approaches towards cyber security. Kor *et al*. [8] on the other hand discussed about security measures in presence of Internet of Things (IoT) integration with mobile applications. They proposed a layered architecture that plays crucial role in security. In a smart healthcare scenario, it was suggested to make use of security measures advised by the law known as HIPAA in distributed environments. Meng *et al*. [9] studied security in terms of "Bring Your Own Device" concept. As the organizations allow devices of users and their smart phones in the work place, they investigated security issues and measures pertaining to detecting malware. They discussed about Android container solutions such as Samsung Knox, secure boot, trusted boot, trustzone, TIMA, sensitive data protection, authentication and access control and security with cryptographic techniques.

In a distributed environment, device to device (D2D) communication is very important. However, such scenario makes use of mobile devices their security concerns may arise. Pedhadiya *et al*. [10] explored different aspects of D2D communication is studied. It includes dynamic situations in vehicular networks, mobile networks, and other networks where distributed scenarios are realized. The study also includes resource allocation related things and also security.

In case of mobile computing and Mobile Cloud Computing (MCC) there is provision for edge computing where local resources are made available to improve response time of the applications. At the same time, it needs security measures to protect applications from cyber-attacks. Xiao *et al*. [6] made review of edge computing and the security measures being taken care of. They discussed different aspects of security. They include weak computation power, attack awareness, OS and protocol heterogeneity, coarse grained access control issues and other security problems. The edge computing architecture is studied along with security measures. They discussed about different attacks such as Distributed Denial of Service (DDoS) and side channel attacks. Malware injection attacks are also discussed in the paper. They used security measures like authentication, authorization, access control and prevention of cyber-attacks.

Edge computing is widely used in IoT as well. Ghorbani and Ahmadzadegan [11] explored security challenges associated with IoT devices that include mobile applications as well. The case studies they discussed include smart homes and other Mobile to Mobile (M2M) environments. The use cases where security is discussed include smart homes, transportation, retail, agriculture, factories and industries, wearable, smart cities and healthcare. Miloslavskaya and Tolstoy [12] also studied IoT environments for study of security. They focused on mobile devices through which malware is spread in such distributed applications. They discussed IoT architecture and the loop holes. There are specific attacks such as DDoS, replay and forgery. Security intelligence is included as part of the infrastructure in order to detect and prevent attacks. It has risk identification and also risk protection. The environment incudes different devices and networks besides applications that cater from simple modules to complex data streaming applications. Security threats may arise due to different reasons. Li *et al*. [13] focused on general hacking approaches that are used by adversaries. It also provides different security issues and countermeasures. Different laws governed by state and international laws are provided. Lee *et al*. [14] proposed a concept known as Secure and Smart Network (S2Net) that is used to protect systems in IoT integrated use cases. It has rich security functionalities. The security features are installed in smart routers where the data is verified and attacks are

detected. Stellios *et al*. [15] on the other hand focused on various attacks that are possible in IoT applications. In such applications Android mobiles are also used. The security provisions take care of application security. From the literature, it is understood that there is need for a malware detection framework based on machine learning for understanding features of. apk files and protect applications in Android devices.

3. PROPOSED MALWARE DETECTION SYSTEM

This section presents the proposed framework and methodology used to detect malware in Android applications. The proposed system takes its data needed from. APK files. It reads. APK files and analyze them and extract different kinds of features. Thus a training dataset is created in order to have better provision for determining the presence of malware associated with. APK file. The proposed system architecture is as shown below.

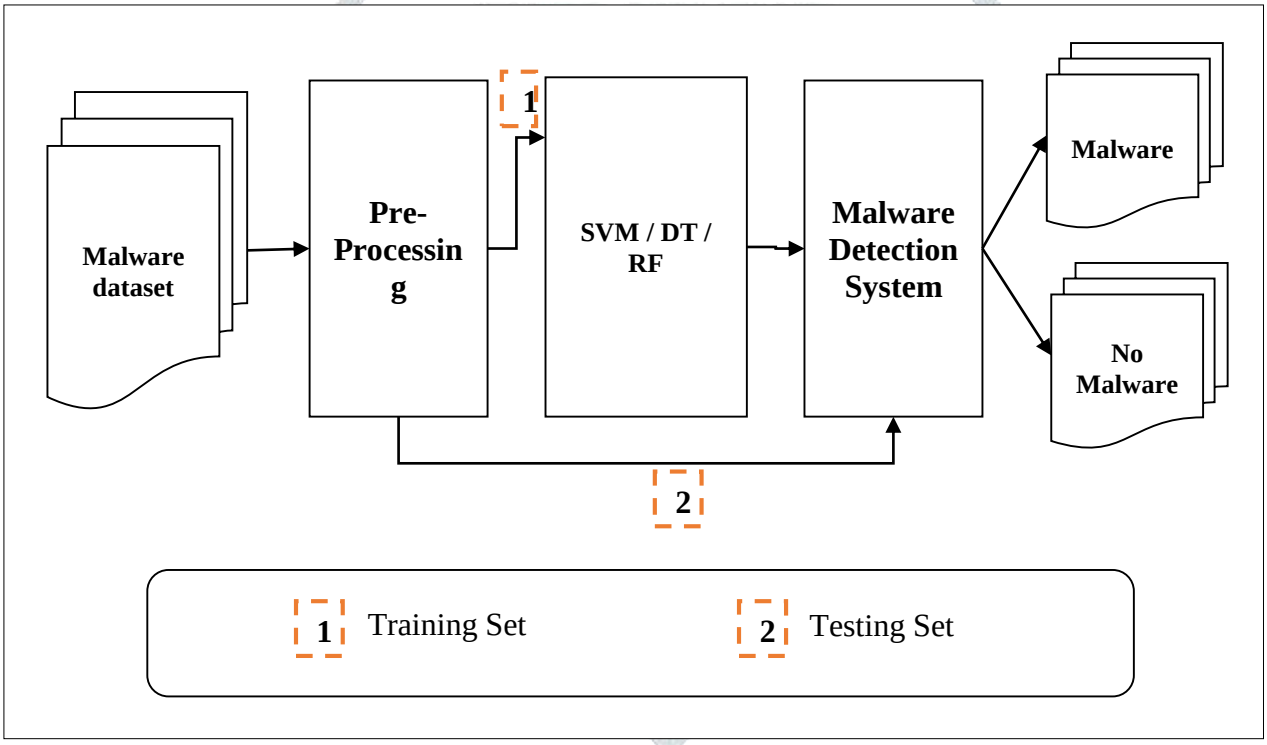

Figure 1: Shows architectural overview of the proposed system

Support vector machine (SVM) is a widely used ML techniques. It analyzes the training set and makes an optimal hyperplane in order to have the capacity to classify the testing samples. The concept of maximum margin is used to clearly distinguish the positive samples from negative ones. SVM is one of the widely used models for malware detection where the optimal hyperplane has power to distinguish classes. It is meant for binary classification by default. However, it can be used for multiple classes with kernels usage. In this project, it is used for binary classification of malware. Multinomial Naïve Bayes is the classifier that is based on multinomial event model. It is the model were events are generated using multinomial. This event model is used for documentation purposes where histogram x is observed as in Eq. 1.

$$P(X\,|C_K) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i pki^{x^i} \qquad (1)$$

Random Forest is used to have multiple decision trees and get ensemble of them with majority voting in order to select the final class label. The RF makes use of many trees from the given instance. The results are ensemble and the majority voting is used in order to finalize the result. the RF chooses training subset. Afterwards, the stop condition is verified. Based on the stop condition, decisions are made to continue operations. With ensemble method, it can handle large volumes of data. It can reuse trees and use them for supervised learning. It is also best used for outlier detection.

Intelligent Malware Detection for Android Applications (IMD-AP)

Algorithm: Intelligent Malware Detection for Android Applications (IMD-AP)

Inputs: APK files, prediction models M

Output: Malware detection results,

1. Start
2. Input malware dataset
3. Pre-processing
4. Extract features from training set (. APK files)
5. For each model m in M
6. Train the model m
7. End For
8. For each model m in M
9. Use model for testing
10. Evaluate
11. Display results
12. End For
13. End

Algorithm 1: Intelligent Malware Detection for Android Applications

As presented in Algorithm 1, it takes APK files and prediction models as input and generate malware detection outputs. It has provision to know the features of each. APK file by analyzing it. It has better possibility to be used in the real time applications. Its accuracy is better than existing system. The Android malware detection models are evaluated using the metrics derived from confusion matrix provided in Table 2.

	Ground Truth (Yes)	Ground Truth (No)
Prediction Model (Yes)	True Positive (TP)	False Positive (FP)
Prediction Model (No)	False Negative (FN)	True Negative (TN)

Table 1: Confusion matrix used to derive different metrics

As presented in Table 1, the TP, FP, TN and FN are used to derive new metrics for performance evaluation. The metrics are provided s in Eq. 1, Eq. 2 and Eq. 3.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F - Measure = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \quad (3)$$

Precision is the measure related to exactness of the detection models while the recall indicates the completeness of the models. The mean of these two is the F-Measure that reflects overall performance of the prediction models.

4. RESULTS AND DISCUSSION

The results are obtained using the performance metrics like precision, recall and F-Measure. Execution time is also compared for all the prediction models used in the empirical study.

Android Malware Detection Model	Performance (%)			
	Precision	Recall	F-Measure	Accuracy
Multinomial NB	0.74704	0.82460	0.77299	0.82399
RF	0.97759	0.94659	0.961498	0.95898
SVM	0.54394	0.54394	0.703874	0.543946

Table 2: Comparison of Android malware prediction models

As presented in Table 2, the prediction models are compared with metrics such as precision, recall and F-measure.

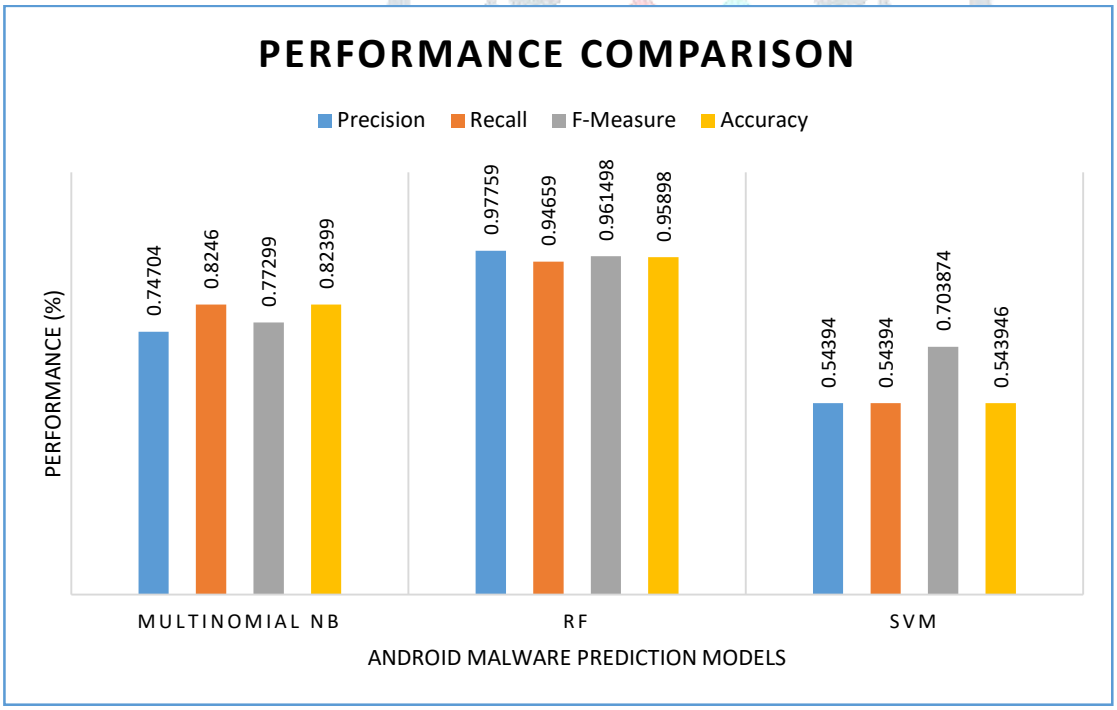

Figure 2: Performance comparison

As presented in Figure 2, the performance of the Android prediction models is compared. The prediction models and the performance (%) are shown in horizontal and vertical axes respectively. The models showed different performance levels. However, RF showed better performance over the other models in terms of accuracy.

Android Malware Prediction Model	Time Taken (seconds)
Multinomial NB	1.45
Random Forest	1.10
SVM	0.37

Table 3: Comparison of execution time (seconds)

As presented in Table 3, the execution time of different Android malware prediction model used in the empirical study are compared.

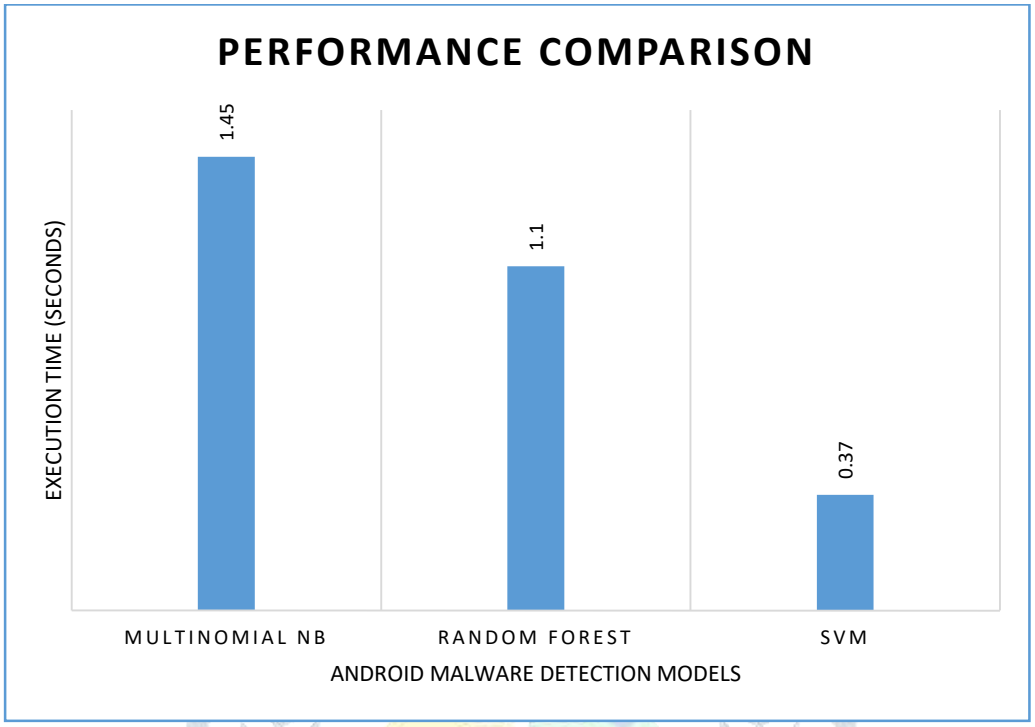

Figure 3: Execution time comparison

As presented in Figure 3, the time taken by each Android malware detection model is shown in vertical axis and the prediction models are provided in horizontal axis. The multinomial NB took 1.45 seconds for completion of Android malware detection. The time taken by the Random Forest model is 1.1 seconds while SVM took 0.37 seconds. Multinomial NB took more time for prediction. SVM showed better performance in terms of execution time.

5. CONCLUSION AND FUTURE WORK

In this project, investigation is made in detection of Android malware. Different supervised learning methods such as Multinomial NB, Random Forest and SVM. Android malware dataset is obtained from. APK files. Different features of the Android applications such as suspicious calls and restricted API calls. Before application of classification models pre-processing is made. Features are extracted from the. APK files. The data is then used for training and prediction purposes. The implementation is made with modularity and the modules are reused in the main source file. The main() method takes command line arguments in order to evaluate prediction models selectively. The results are evaluated using metrics like precision, recall and F-measure besides execution time. The results revealed that RF showed 0.95898 accuracy which is better than multinomial NB and SVM that show 0.82399 and 0.543946 respectively. With respect to execution time, the SVM model showed better performance in terms of execution time.

References

[1] Anwar, S., Zain, J. M., Inayat, Z., Haq, R. U., Karim, A., & Jabir, A. N. (2016). A static approach towards mobile botnet detection. 2016 3rd International Conference on Electronic Design (ICED) p1-5

[2] Hatcher, W. G., & Yu, W. (2018). A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends. IEEE Access, 6, p24411–24432

[3] Hasan, R., Zawoad, S., & Haque, M. M. (2016). StuxMob: A situational-aware malware for targeted attack on smart mobile devices. MILCOM 2016 - 2016 IEEE Military Communications Conference p1-6

[4] Han, Q., Liang, S., & Zhang, H. (2015). Mobile cloud sensing, big data, and 5G networks make an intelligent and smart world. IEEE Network, 29(2),p 40–45

[5] Malatras, A., Freyssinet, E., & Beslay, L. (2015). Mobile Botnets Taxonomy and Challenges. 2015 European Intelligence and Security Informatics Conference p1-4

[6] Xiao, Y., Jia, Y., Liu, C., Cheng, X., Yu, J., & Lv, W. (2019). Edge Computing Security: State of the Art and Challenges. Proceedings of the IEEE, p1–24.

[7] Eustace, K., Islam, R., Tsang, P., & Fellows, G. (2018). Human Factors, Self-awareness and Intervention Approaches in Cyber Security When Using Mobile Devices and Social Networks. Security and Privacy in Communication Networks, p166–181

[8] Kor, A.-L., Yanovsky, M., Pattinson, C., & Kharchenko, V. (2016). SMART-ITEM: IoT-enabled smart living. 2016 Future Technologies Conference (FTC)p1-11

[9] Meng, T., Shang, Z., & Wolter, K. (2017). An empirical performance and security evaluation of android container solutions. 2017 International Conference on Cyber Security And Protection Of Digital Services (Cyber Security).p1-8

[10]Pedhadiya, M. K., Jha, R. K., & Bhatt, H. G. (2018). Device to device communication: A survey. Journal of Network and Computer Applications.p1-26

[11] Ghorbani, H. R., & Ahmadzadegan, M. H. (2017). Security challenges in internet of things: survey. 2017 IEEE Conference on Wireless Sensors (ICWiSe).p1-6

[12] Miloslavskaya, N., & Tolstoy, A. (2017). Ensuring Information Security for Internet of Things. 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud).p 1-8

[13] Li, C.-Y., Huang, C.-C., Lai, F., Lee, S.-L., & Wu, J. (2018). A Comprehensive Overview of Government Hacking Worldwide. IEEE Access, 1–21.

[14] Lee, S., Shi, H., Tan, K., Liu, Y., Lee, S., & Cui, Y. (2019). S2Net: Preserving Privacy in Smart Home Routers. IEEE Transactions on Dependable and Secure Computing, p1–13

[15] Stellios, I., Kotzanikolaou, P., Psarakis, M., Alcaraz, C., & Lopez, J. (2018). A Survey of IoT-enabled Cyberattacks: Assessing Attack Paths to Critical Infrastructures and Services. IEEE Communications Surveys & Tutorials, 1–43