International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 Impact Factor: 8.176 ISSN: 2582-3930

ANOMALY DETECTION IN WIRELESS DEVICES USING CHANGE POINT ANALYSIS

N Kumaran,
Assistant Professor, Dept. Of CSE, SCSVMV University, TN, India
E-mail: nkumaran@kanchiuniv.ac.in

Peteti Mukesh,
B.E, Final Year Student, Dept. Of CSE, SCSVMV University, TN, India
E-mail: pmukeshwork@gmail.com

Samudrala Ajay kumar,
B. E, Final Year Student, Dept. Of CSE, SCSVMV University, TN, India
E-mail: 11199a212@kanchiuniv.ac.in

Abstract – To give end consumers a trustworthy experience, smartphones and IoT (Internet of Things) devices now need to have three key qualities: security, reliability, and availability. These qualities can be diminished by unrelated incidents or strange conduct that results in hardware damage, software changes, the theft of user information, and a negative influence on the speed or availability of the device. In light of these realities, this project focuses on exploiting smartphone power usage signals for anomaly identification. These signals indicate the device's dynamic behaviour as a result of the interaction of various hardware elements that are simultaneously controlled by one or more applications.

Due to the variations in its statistical features over time, this behaviour can be viewed as a non-stationary process. In light of this presumption, our methodology applies a changepoint detection theory-based feature extraction strategy. Then, it integrates three machine learning classifiers to maximise detection performance and add diversity. A dataset of simulated malware that was running in the background on a smartphone was used to validate the methods. AdaBoost, XGBoost, decision trees, and support classifiers have all been employed. The correctness of the provided data set is examined using these four key algorithms.

Key words – Anomalous Behaviour, AdaBoost Classifiers, Decision tree, and Support Vector Classifier XGBoosting.

1. INTRODUCTION

The number of connected and future-ready gadgets has exploded in the Internet of Things (IoT) sector in recent years. The number of devices is predicted to reach 50 billion by 2020. These gadgets make up a sizable portion of smartphones. In 2019, there are about 2.7 billion smartphone users worldwide, according to Statista. The constant requirement for Internet connection via wireless communication systems to send sensitive and private information of users is a typical feature of IoT

devices and smartphones. Cybercriminals who have created programmes with malicious code to steal passwords, emails, contacts, images, videos, health insights, or other valuable user information have taken an interest in these devices as a result. Other cybercriminals have concentrated their efforts on cellular, private, and public network infrastructure degradation and harm, turning IoT devices into botnets to cause denial of service attacks on these networks.

Researchers, businesses, and even governments have been creating various ways to stop such unwanted acts. The majority of these methods are based on scrutinising the static properties of the source code of the applications. This tactic's vulnerability to code modification and obfuscation in an effort to escape detection is a drawback. As a result, other researchers have been creating approaches for analysing the dynamic properties of the device, such as network traffic, power consumption, Central Processing Unit (CPU) activity, and temperature when applications are operating. Both real-time (online) and offline (analysis of measures acquired beforehand) options are available for this investigation. To perform more accurate recognition, some researchers have employed hybrid approaches, which combine static and dynamic properties.

The current work suggests a fresh way for determining whether a smartphone is running a malicious application by looking at how much power the device is utilising. The idea is that a device's power consumption contains useful information that is encoded and may be utilised to detect the existence of malware. This is because when malware is installed in a device, it is required to carry out certain tasks that represent the total energy used by all of the hardware components of the device, including the CPU, network components, screen, Global Positioning System (GPS), accelerometers, and other components. This methodology assumes that significant features may be incorporated in very short amounts of time and uses offline processing methods and off-device measurement, which requires an external device to collect the power usage. Furthermore, we extract features from a non-stationary time series signal using the notion of changepoint detection.

2. LITERATURE SURVEY

Based on: D. Evans, “The internet of things: How the next evolution of the internet is changing everything,” CISCO white paper, Tech. Rep., 2011.

There are many middleware solutions for the Internet of Things available today, enabling the connectivity of the sensors and actuators. To be widely adopted, these solutions—referred to as platforms—must live up to the requirements of various IoT ecosystem participants, including devices.

The Internet of Things (IoT), often known as low-cost devices that can connect wirelessly to the Internet, includes everything from smartphones to coffee makers. The technique and development process for building an IoT platform are described in this study. The architecture and implementation for the IoT platform are also presented in this paper. The objective of this project is to create an analytics engine that can collect sensor data from various devices, give users the ability to extract useful information from IoT data, and then use that information to take action using machine learning algorithms. To enhance system performance overall and enable simple scalability, the suggested system is introducing the usage of a message system.

Statista, "Number of mobile phone users worldwide from 2015 to 2020 (in billions)." [Online]. Available: https://www.statista.com/statistics/274774/forecast-ofmobile-phone-users-worldwide/.

The development of virtual reality has brought augmented reality (AR) to a new level of perspectives for what it means to see, hear, and be immersed in the actual world. In the past ten years, the development of mobile devices has made augmented reality (AR) a cutting-edge technology, giving rise to an increasing number of location-based mobile AR (LBMAR) systems. There are, however, just a small number of review studies that have concentrated on examining elements like growth, types, characteristics, features, sensors, application fields, and their corresponding difficulties. In this work, a thorough analysis of location-based mobile augmented reality (LBMAR) systems is presented. 35 papers in total published between 2013 and 2018 in the top six most visited indexed databases are examined. The systematic review was carried out using the Kitchenham approach, and the PRISMA method was used to analyse the results. The LBMAR system is given a thorough review in this chapter, along with a list of the research questions that still need to be answered.

A. Arabo and B. Pranggono, "Mobile malware and smart device security: Trends, challenges and solutions," in 2013 19th International Conference on Control Systems and Computer Science, May 2013, pp. 526–531.

conditions and environments. It can be challenging to monitor and comprehend changes to modules and relationships inside a software project, and this difficulty increases as the software undergoes multiple changes. It is also more difficult to comprehend software evolution patterns due to the normal complexity and scale of software. In this study, we provide an animated interactive matrix-based visualisation method that shows the evolution of software ideas. It displays, for instance, which

new couplings and modules are added and withdrawn over time. Our general visualisation, which is used in the context of software evolution, supports dynamic and weighted digraphs. The structural organisation of the software can be ascertained by examining source code modifications, which also help to spot quality problems over time. We investigate open-source repositories to illustrate our method and talk about some of our findings on these changing software architectures.

This effort is a component of the investigation of the issues and trends in cyber security for smart homes and smart devices. We have observed the growth and need for the seamless interconnection of smart devices to offer users a range of capabilities and capacities. These gadgets give users greater capabilities and usefulness, but they also present new risks and dangers. The discussion and analysis of current smart device cyber security challenges follows. The context and motive for the paper are discussed at the outset. One of the biggest problems with the security of smart devices, according to us, is mobile malware. Users of mobile smart devices should anticipate a dramatic rise in malware and significant developments in malware-related assaults in the near future, particularly on the Android platform given its rapidly expanding user base. We go into great detail about and study mobile malware, identifying issues and potential future developments. Then, in order to address the problem, we suggest and talk about an integrated security solution for smart device cyber security. incentive, too.

T. Kim, B. Kang, M. Rho, and et. all, “A multimodal deep learning method for android malware detection using various features,” IEEE Trans. on Info. Forensics and Security, vol. 14, no. 3, 2019.

The prevalence of cell phones has significantly increased the amount of malwares. Because of their widespread use, Android devices are among the smart gadgets that malware targets the most. The unique framework for Android malware detection is proposed in this study. For effective feature representation on malware detection, our framework employs a variety of features to reflect the characteristics of Android applications from multiple angles. The features are enhanced using our existence-based or similarity-based feature extraction approach. A multimodal deep learning approach is additionally suggested for usage as a malware detection model. This study using multimodal deep learning for Android malware detection is the first of its kind. We were able to maximise the advantages of including several feature types with our detection model. We ran a number of trials with a total of 41,260 samples to assess the performance. We contrasted our model's precision with other deep neural network models'. Additionally, we assessed the effectiveness of model updates, the value of various features, and our feature representation approach when it came to our framework. We also evaluated

how well our framework performed in comparison to other approaches, including ones that relied on deep learning.

3. PROPOSED SYSTEM

Several machine learning models were put forth in this system to categorise whether or not there is a foodborne illness, but none have sufficiently addressed the issue of misdiagnosis. Additionally, comparable studies that have presented models for evaluating this performance classification frequently ignore the heterogeneity and amount of the data. So, as a means of prediction, we suggest a Random Forest, Decision Tree, Gradient Boosting, and AdaBoost Classifier.

3.1 System Architecture

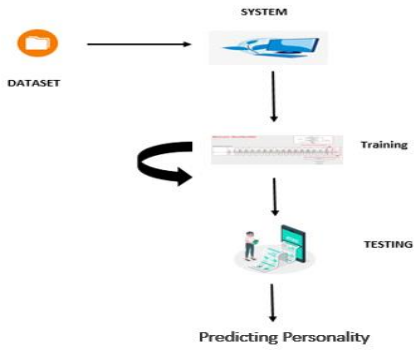

Fig.1. System Architecture

3.2 Algorithm

3.2.1 AdaBoost Classifier:

Machine learning ensemble methods use the boosting technique known as the AdaBoost algorithm, sometimes referred to as Adaptive Boosting. Since the weights are reallocated to each instance, instances that were incorrectly classified are given higher weights, hence the term "adaptive boosting." Boosting is used to reduce bias and variation in supervised learning. It is based on the idea that children make progress in stages. Except for the first, every student after that is created from a previous learner. Simply put, weak students become strong students. Boosting is conceptually similar to the AdaBoost approach, however it is slightly different. Let's go into more detail about this distinction.

Let's first talk about how boosting functions. During the data training phase, 'n' decision trees are created. The improperly classified record in the first model is given priority as the first decision tree or model is constructed. For the second model, just these records are sent as input. The procedure continues

until we decide how many base learners to produce. Remember that all boosting approaches permit record repetition.

3.2.2 Decision Tree Classifier:

In addition to having many applications in daily life, trees have an impact on both classification and regression in a variety of machine learning contexts. In decision analysis, a decision tree can be used to formally and graphically represent decisions and decision-making. As implied by the name, it uses a decision-tree-like methodology. Despite being a regularly used technique in data mining for creating a plan to accomplish a particular goal.

An upside-down decision tree is depicted, with the root at the top. In the left figure, the bold writing in black denotes an internal node or condition upon which the tree's branches and edges are based. The choice or leaf, in this case, whether the passenger died or lived, is shown as red and green text, respectively, at the end of the branch that doesn't divide any longer.

3.2.3 XGBoosting Classifier:

"Extreme Gradient Boosting" is the abbreviation for XGBoost. XGBoost is a distributed gradient boosting library that has been optimised to be very effective, adaptable, and portable. It uses the Gradient Boosting framework to implement machine learning algorithms. It offers a parallel tree boosting to quickly and accurately address a variety of data science challenges.

4. METHODOLOGY

4.1 Modules:

4.1.1 User:

See the home page

The user views the web application's main page for abnormal behaviours.

Visit the Upload page

Users can submit the emulated data from the smartphone device and read more about the prediction of anomalous behaviour on the about page.

Enter Model

In order to receive results, the user must enter values into specific fields.

Results View

The generated results from the model are displayed to the user, who may see whether any anomalous activity exists or not. using machine learning.

View rating

By applying the four primary algorithms, the user has the option to view the score in percent of the data accuracy.

4.1.2 System:

preparing a dataset

The system scans for data, loads it into CSV files, and determines if it is available or not.

Pre-processing

Data must be pre-processed in accordance with the models in order to improve the model's correctness and the data's knowledge.

developing the data

Before training with the provided methods, the pre-processed data will be divided into train and test halves.

Building a model

This lesson will assist in developing a model that predicts personality more accurately.

Produced Score

User can view the score in % here.

Produce Results:

Our machine learning algorithm is trained, and it makes predictions about unusual behaviours.

5. PROJECT DESCRIPTION

A fresh approach to spotting odd behaviour in mobile applications. The study takes into account the use of three machine learning approaches to train a classifier from the power used by the smartphone and the use of a changepoint detection theory to extract features. Comparing the suggested technique to the other three methodologies, we can draw the conclusion that it performs better in terms of F1-measure accuracy. We want to emphasise that one advantage of our methodology over other approaches is that it can detect malware that acts quickly. Instead of utilising an emulated malware, we want to extend the concept to real malware in future work.

6. RESULT

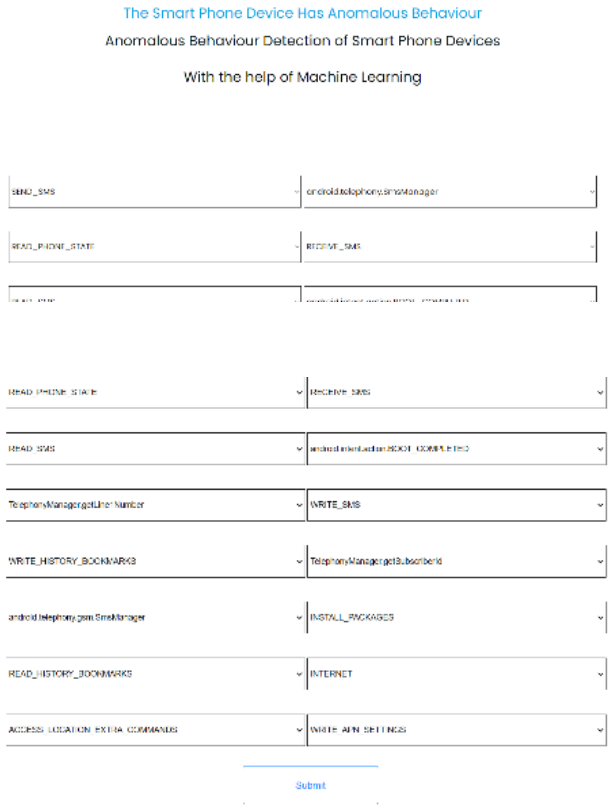

Fig.1. Output

S. No	Algorithms Used	Accuracy	Time Taken
1.	Decision Tree Classifier	82.04	28s
2.	Support Vector Classifier	89.45	19s

Table-1(Existing Algorithms)

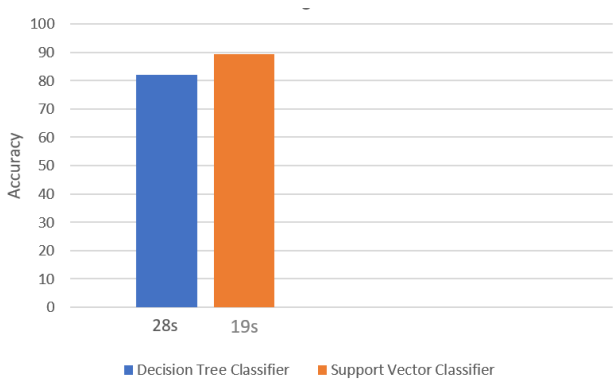

Chart-1(Existing Algorithms)

S. No	Algorithms Used	Accuracy	Time Taken
1.	Decision Tree Classifier	90.37	1.5s
2.	Support Vector Classifier	83.28	3s
3.	XGBoost Classifier	90.25	1s
4.	AdaBoost Classifier	83.57	1s

Table-2(Proposed Algorithms)

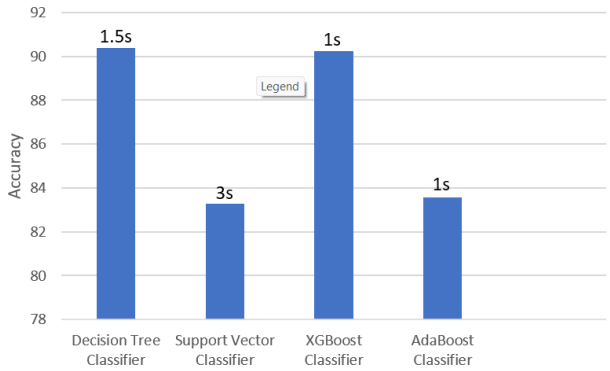

Chart-2(Proposed Algorithms)

7. CONCLUSION

A fresh approach was put forth to spot odd behaviour in smartphone applications. The study takes into account the use of three machine learning approaches to train a classifier from the power used by the smartphone and the use of a changepoint detection theory to extract features. Comparing the suggested technique to the other three methodologies, we can draw the conclusion that it performs better in terms of F1-measure accuracy. We want to emphasise that one advantage of our methodology over other approaches is that it can detect malware that acts quickly.

Instead of utilising an emulated malware, we want to extend the concept to real malware in future work.

REFERENCES

[1] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," CISCO white paper, Tech. Rep., 2011.

[2] Statista, "Number of mobile phone users worldwide from 2015 to 2020 (in billions)." [Online]. Available: https://www.statista.com/statistics/274774/forecast-ofmobile-phone-users-worldwide/

[3] A. Arabo and B. Pranggono, "Mobile malware and smart device security: Trends, challenges and solutions," in 2013 19th International Conference on Control Systems and Computer Science, May 2013, pp. 526–531.

[4] T. Kim, B. Kang, M. Rho, and et. all, "A multimodal deep learning method for android malware detection using various features," IEEE Trans. on Info. Forensics and Security, vol. 14, no. 3, 2019.

[5] Y.-S. Yen and H.-M. Sun, "An android mutation malware detection based on deep learning using visualization of importance from codes," Microelectronics Reliability, vol. 93, pp. 109–114, 2019.

[6] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. Siemens, "Drebin: Effective and explainable detection of android malware in your pocket." in Ndss, vol. 14, 2014, pp. 23–26.

[7] P. Faruki, A. Bharmal, V. Laxmi, and et. all, "Android security: A survey of issues, malware penetration, and defenses," IEEE Communications Surveys Tutorials, vol. 17, no. 2, pp. 998–1022, Secondquarter 2015.

[8] K. Ariyapala, H. G. Do, H. N. Anh, and et. all, "A host and network based intrusion detection for android smartphones," in 30th Int. Conf. on Advanced Info. Net. and Apps Workshops (WAINA), March 2016.

[9] M. Curti, A. Merlo, M. Migliardi, and S. Schiappacasse, “Towards energy-aware intrusion detection systems on mobile devices,” in Int. Conf. on High Performance Computing Simulation (HPCS), July 2013.

[10] H. Kim, J. Smith, and K. G. Shin, “Detecting energy-greedy anomalies and mobile malware variants,” in Proceedings of the 6th international conference on Mobile systems, applications, and services. ACM, 2008.

[11] H. Kurniawan, Y. Rosmansyah, and B. Dabarsyah, “Android anomaly detection system using machine learning classification,” in Int. Conf. on Electrical Engineering and Informatics (ICEEI). IEEE, 2015.

[12] T. Zefferer, P. Teufl, D. Derler, and et. all, “Towards secure mobile computing: Employing power-consumption information to detect malware on mobile devices,” Int. journal on advances in software, vol. 7, 2014.

[13] R. James, A. Albasir, K. Naik, M. Y. Dabbagh, and et. all, “Detection of anomalous behavior of smartphones using signal processing and machine learning techniques,” in 2017 IEEE 28th Annual Int. Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017.

[14] R. James, A. Albasir, K. Naik, and et. all, “A power signal based dynamic approach to detecting anomalous behavior in wireless devices,” in Proceedings of the 16th ACM Int. Symposium on Mobility Management and Wireless Access MobiWac’18, 2018.

[15] L. Al Shalabi, Z. Shaaban, and B. Kasasbeh, “Data mining: A preprocessing engine,” Journal of Computer Science, vol. 2, no. 9, 2006.