International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5899
***Article Received**: 12 October 2022 **Revised**: 12 November 2022 **Accepted**: 12 December 2022*

A Hybrid Model for Android Malware Detection using Decision Tree and KNN

Heena Kauser.Sk[1], Dr. Maria Anu.V[2]
[1]Research scholar,
Department of Computer Science & Engineering,
Sathyabama Institute of Science and Technology,
Chennai, Tamil Nadu, India.
e-mail: heenacse2020@gmail.com
[2]Professor,
Department of Computer Science & Engineering,
Sathyabama Institute of Science and Technology,
Chennai, Tamil Nadu, India.
e-mail: mariaanu18@gmail.com

Abstract— Malwares are becoming a major problem nowadays all around the world in android operating systems. The malware is a piece of software developed for harming or exploiting certain other hardware as well as software. The term Malware is also known as malicious software which is utilized to define Trojans, viruses, as well as other kinds of spyware. There have been developed many kinds of techniques for protecting the android operating systems from malware during the last decade. However, the existing techniques have numerous drawbacks such as accuracy to detect the type of malware in real-time in a quick manner for protecting the android operating systems. In this article, the authors developed a hybrid model for android malware detection using a decision tree and KNN (k-nearest neighbours) technique. First, Dalvik opcode, as well as real opcode, was pulled out by using the reverse procedure of the android software. Secondly, eigenvectors of sampling were produced by utilizing the n-gram model. Our suggested hybrid model efficiently combines KNN along with the decision tree for effective detection of the android malware in real-time. The outcome of the proposed scheme illustrates that the proposed hybrid model is better in terms of the accurate detection of any kind of malware from the Android operating system in a fast and accurate manner. In this experiment, 815 sample size was selected for the normal samples and the 3268-sample size was selected for the malicious samples. Our proposed hybrid model provides pragmatic values of the parameters namely precision, ACC along with the Recall, and F1 such as 0.93, 0.98, 0.96, and 0.99 along with 0.94, 0.99, 0.93, and 0.99 respectively. In the future, there are vital possibilities to carry out more research in this field to develop new methods for Android malware detection.

Keywords- Android Malware, Decision Tree, KNN, Precision, Recall, Smartphone.

I. INTRODUCTION

Mobile internet users are increasing day by day all around the world. With the Android operating system, smartphones are continuously becoming the essential gadget in the lives of people across the globe [1]. By the last of 2023, mobiles adoption among individuals across advanced economies would approach 90.00%, up from 85.00% in the year 2018, whereas worldwide cell phone shipments would exceed 1.85 billion handsets, up 19.00% from the year 2018 [2]. Presently, the most popular operating systems (OS) for cell phones seem to be iOS as well as Android, along with the Windows Phone. But, nowadays Android-based operating system, particularly, has become one of the most popular OS (operating systems) along with the largest marketplace share on a world wide spectrum due to its accessible environment, which also allows consumers as well as programmers to customize basic features [3], [4]. As per the poll findings given by Gartner, an Android system had 85.90% market dominance throughout 2017. The growing adoption of Android seems to be very high, conversely, associated with an increase in spyware [4].

Smartphones are becoming a primary mode of communication with each other using voice calls, text messages, and another app-based chatting. The first cell phone was a standard mobile handset, but with substantial technological advancements, has indeed evolved into something like a Smartphone [5]–[7]. Due to the accessible features, the quantity of Smartphones consumers is higher than before. Previously, a cell phone might have been utilized to initiate phone calls as well as send text messaging. Smartphones are now being utilized as cameras, audio players, iPad, as well as web browsers, among other things. Smartphones nowadays are outfitted with many detectors, as well as increased storage and computing capability, allowing smartphones to function as a complete computing devices [8]–[10]. Every month, the volume of ransomware grows at even a quicker pace, posing a severe safety issue; antivirus providers

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5899
***Article Received**: 12 October 2022 **Revised**: 12 November 2022 **Accepted**: 12 December 2022*

discover hundreds of different malicious specimens every day, with really no limit in view [11], [12]. With the steady maturation of 5G networks, that heralds the dawn of an age of sophisticated networking as well as mobile IoT (Internet of Things), the Web of Anything would see ransomware become increasingly dangerous as well as widespread, making spyware identification a vital problem in cybersecurity [13]–[16]. Figure 1 illustrates the diverse types of malware classification. A few of the common malware are viruses, botnets, spyware as well as ransomware.

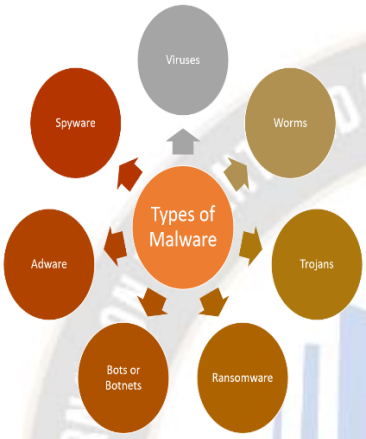

Fig. 1: Illustrates the diverse types of malware classification. A few of the common malware are viruses, botnets, spyware as well as ransomware.

Because of the widespread adoption of cell phones, each resident is now at risk of receiving undesired as well as dangerous apps. With the growing quantity of Android customers, spyware programmers are continuously building harmful apps. According to a previous study, malware ELF (Executable and Linkable Format) binaries repackage Android applications to hide calls to outside libraries [17]–[20]. Furthermore, experts are attempting to determine the most effective spyware identification techniques, such as memory forensic techniques as well as safety data transfer methodologies, for Android smartphones. While a customer wishes to acquire any app through the Google-based Play Store, the android app will be initially retrieved, and afterward, the customer is requested to apply it following agreeing on all rights. A person could indeed install any app unless they agree to all of the designer's authorization [21]–[23]. Hackers frequently request authorization to examine a user's webcam, voice, texts, and emails, as well as other personal data. Customers are unaware of the hackers' intent therefore allow all rights to load the specific app as per the choice. As a result, individuals now become a target of a scammer's assault. Attackers could also alter persistent strings to launch attacks on android platforms [24]–[26].

1.1. Research Contribution:

- In this article, the authors developed a novel hybrid and effective model for android malware detection using a combination of the decision tree and the KNN technique.
- Our hybrid model is capable of identifying the multiple types of malwares such as the spyware, ransomware, and diverse viruses along with the trojans in less time and an accurate manner, and based upon that capable to detect in the android based operating platform for providing the additional secrecy.
- The accuracy and performance of the proposed model are improved and considerable in comparison to the existing schemes developed by investigators during the last decade.

This experiment has been carried out with high precision and accuracy for results measurement. Further, the proposed hybrid model is also compared with the existing CNN (Convolutional Neural Network) for verification of the proposed model in a pragmatic manner and measured results are found enhanced with the proposed hybrid model.

II. Literature Review

Y. Pan et al. in [27], discussed existing techniques of the android malicious detection in their literature article. Because of the ubiquitousness of the Android-based OS (operating system), Android-based malicious issues are increasing constantly these days all around the world. been on the rise in recent years. Android-based Spyware is executed on cell phones outside the consumers' knowledge or consent, as well as it brings serious risks to consumers, including the theft of sensitive data as well as technological theft. Scientists, as well as professionals, have offered a variety of strategies to counter such dangers. One of such approaches involves static analysis, which is frequently used within Android-based malware identification and may identify malware rapidly as well as prevent it from being installed. C. Li et al. [28], proposed another method of android malware identification which is rooted in the factorization machine. The android-based mobiles are becoming more popular all around the world due to their pragmatic execution speed along with the less computational complexity as well as the overall cost of maintenance is lower and more reliable in comparison to other operating systems. The malicious assaults are increasing day by day on the android based operating systems. The conventional schemes like signature-rooted routines seem to be incapable of providing the required secrecy to the confidential datasets. However, this existing method is not providing the required secrecy in the upgraded versions of the android-based operating systems.

K. Liu et al. in [29], discussed another review article on the topic of android-rooted malicious identification schemes rooted in the machine learning techniques. Android-

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5899
***Article Received**: 12 October 2022 **Revised**: 12 November 2022 **Accepted**: 12 December 2022*

based apps are fast evolving throughout the smartphone environment, yet Android ransomware is always appearing. Several investigators have looked at the issue of Android spyware identification as well as proposed hypotheses and approaches through various angles. Machine learning techniques appear to be an efficient as well as an interesting method for detecting Malicious activity, according to previous studies. However, there have already been evaluations that have looked into various aspects of Android spyware identification using machine learning. Through evaluating a broader variety of facets of this issue, researchers feel new work enhances prior investigations. This article gives a thorough assessment of machine learning-based Android spyware identification techniques.

S. K. Sasidharan et al. in [30], discussed another android-based malicious identification scheme that is rooted in the outline secreted Markov model. Because of its accessibility as well as flexibility, the Android operating system has become a possible victim of malicious assaults. Hackers are constantly refining as well as improving their assault tactics to find weaknesses in updated Android editions. Because of the safety constraints as well as resources limits inherent in such smartphones, detecting as well as analyzing malicious assaults on the Android operating system presents unusual hurdles. This study presents a novel behavioral technique for detecting as well as classifying Mobile spyware. This scheme is capable of identifying the unknown apps as benign as well as malware rooted over log probability score produced. However, this approach has numerous drawbacks such as high computational complexity as well as incapable to operate on the higher android version in a pragmatic manner. H. Chen et al. in [31], discussed another scheme of the android malicious activities identification utilizing the agreement-sensitive features. Android's accessibility has rendered it a favorite of both users as well as programmers, resulting in a surge in smartphone applications. However, its prominence draws the notice of potential assailants. Android ransomware continues to pose a threat to consumers' confidentiality. As a result, developing a rigorous as well as adaptable solution for Android spyware identification is useful. This article demonstrates how to construct a virus identification solution for Android using a structured methodology.

W. Zhang et al. in [32], proposed another scheme for the android malicious activities identification by utilizing the bytecode pictures method as well as TCN (temporal convolution). The pictures-rooted analytical approach is becoming an efficient means to guard against symmetric cryptography as well as misleading spyware, due to the fast growth in the amount of Android malicious software. Currently, the convolution neural network (CNN)-based Ransomware bytecode picture identification approach depends on a single particular DEX file characteristic, therefore, taking a significant quantity of processing. The major drawback of this scheme is lower performance constraints such as Recall 95.45 percent and the F1-score was measured at 95.44 percent only.

III. METHODOLOGY

Now-a-days effective malware detection has become one of the significant factors in the secrecy of android operating systems. However, the existing used signature-rooted approaches could not offer correct recognition of real-time malware assaults as well as polymorphic viruses. In this article, for resolving the aforementioned drawbacks, the authors developed a hybrid model which is rooted in the combination of the decision tree as well as the KNN technique.

3.1. Proposed Hybrid Model based on Decision Tree and KNN:

Our proposed scheme utilizes the combination of the heuristic rooted analysis as well as the signature method for the android-based applications. In this approach, authors converse engineered android operated applications for the extraction of the manifest files of varied sizes from the archives and binaries as well as employed KNN along with the decision tree algorithm for effective detection of the malware in real-time. For achieving the effective detection of the malware in real-time, multiple experiments have been conducted utilizing the multiple classifiers namely the Decision Tree as well as KNN algorithms jointly for performance measurement of our suggested model. Another advantage of the suggested model is eliminating the chances of overfitting as well as the weaker generalization capability within the decision tree algorithm by combining the decision tree along with the KNN technique.

The KNN algorithm was integrated with the decision tree for the accurate optimization of the nodes that confirms higher accurateness of the decision pathway as well as enhances the generalization capability of the decision tree. Further, our suggested scheme is capable of performing the large as well as little samples training effectively for malware detection in the android-based applications. Figure 2 illustrates the proposed hybrid model for android malware detection by a combination of the decision tree along with the KNN technique. Our suggested developed hybrid model is segregated into the five diverse segments which are sample set and extract opcode along with the features engineering as well as decision tree-KNN and the last one is test set verification. The objective of the suggested hybrid model was to execute the combined decision tree and the KNN algorithm for the generation of the decision tree rooted over the sampling dataset as well as

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5899
***Article Received**: 12 October 2022 **Revised**: 12 November 2022 **Accepted**: 12 December 2022*

further update all of the decision nodes from the bottom side to the upside in an effective manner.

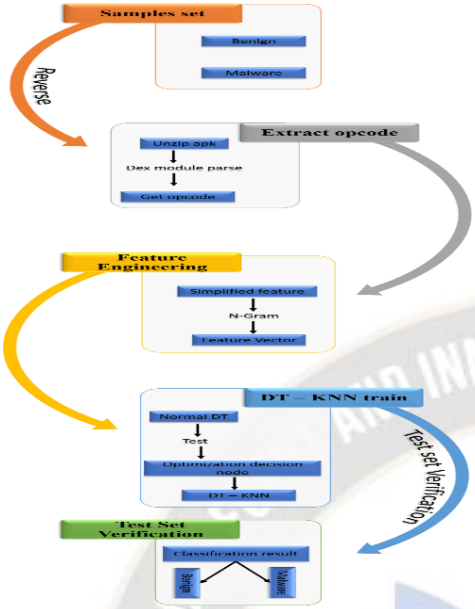

Fig. 2: Illustrates the proposed hybrid model for android malware detection by a combination of the decision tree along with the KNN technique. Our suggested developed hybrid model is segregated into the five diverse segments which are sample set and extract opcode along with the features engineering as well as decision tree-KNN and the last one is test set verification.

3.2. Instrument:

This experimentation was performed utilizing the unzip, as well as dexParser along with the scikit-learn. The scikit-learn is known for the outstanding python coding ML-based (machine-learning) library. This library is configured with various types of classification, as well as clustering algorithms which include the KNN, vector machine. This experiment was performed on a computer system having the following system configuration: Solid State Drive (SSD) 512 GB, RAM (Random Access Memory) 16 GB, Intel i5 processor with the 64-bit operating system.

3.3. KNN with the Decision Tree:

In this article, the KNN as well as the Decision Tree algorithm-based hybrid model has been developed for overcoming the threats of overfitting as well as weaker generalization capability in the earlier developed algorithms such as in the SVM (Support Vector Machine). The decision tree is the class of the supervised learning-based algorithm. The decision tree is known as general classification as well as a regression technique that categorizes the chosen sample in a tree kind of shape. The decision tree shows a procedure to classify samples rooted in features in categorization issues as well as recognized as a gathering of if-then set of rules. The decision tree has been utilized globally due to the intuitive features definition, higher categorization accurateness, along with easy implementation procedure in real-time. The decision tree algorithm learning procedure is quite easy which is to obtain a mapping relation between items features as well as items amount. Further, the decision tree allows for the generalization of multiple categorization rules illustrated through tree shapes from arbitrary samples. KNN algorithm was selected with the decision tree due to less computational complexity and easy implementation procedure in real-time. Further, the KNN method is a pragmatic solution for resolving the classification as well as the regression issues in a very less period.

3.4. System Initialization Procedure:

The decision tree is the classification of the supervised learning algorithm. In the proposed hybrid model which is based on the decision tree and the KNN technique, the entire sampling set represented by the Sa = $\{(p_1, q_1)$, $((p_2, q_2)$, $((p_3, q_3)$,… $(p_n, q_n)\}$ was segregated in the training group as well as the testing group which is represented by the TrSet as well as TxSet.

Step 1: As per the TrSet, Gini indexing was utilized to select features as well as prepruning, and based upon that decision tree was constructed.

Step 2: Used TxSet for evaluation of decision tree as well as evaluated precision of entire decision pathway PXi, after that, create decision item do = (Cdpi, CPi, Chi), further setting of the decision pathway accurateness threshold value CTh.

Step 3: Initiate train CQ = {}, tree decisions sorting along with the items generation in the previous step in a downward direction as per the pathway depth that is Cdh of decision pathway Cdp, further in a series addition of each item in the defined train CQ = {}.

Step 4: To evaluate whether the train is unfilled. While the train CQ = {} found unfilled, defined algorithms terminate. Or else, go to the next step.

Step 5: Fetching components Cq = (Cdp, CP, Ch) from the train, further comparison of decision pathway precision frequency CXp with a predefined threshold value that is CTh. While it is found below the CTh, go to the next step. Or else, go to previous step 4 for retaining the decision pathway.

Step 6: To evaluate entire sibling nodes of the train CQ = {} for evaluation of the leaf nodes. While leaf node detected go to next step. Or else, go ahead to next step 8 immediately.

Step 7: To evaluate pathway precision of the train CQ = {}.

Step 8: Go ahead with taking entire trained datasets of pathway CP and training along with the KNN algorithm, further update the entire KNN nodes.

Step 9: After that go ahead, again to the previous fourth step as well as remain for traversing for updating the nodes.

Step 10: End.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5899
***Article Received**: 12 October 2022 **Revised**: 12 November 2022 **Accepted**: 12 December 2022*

IV. Results And Discussion

In this article, the authors developed a novel hybrid model which is based on the decision tree and the KNN technique. It was found after the experiment that the suggested hybrid model using the KNN along with the Decision Tree algorithm is capable to manifest varied sizes of the .xml files and is one of the accurate solutions to detect the malware found in the android-based applications. Thus, the suggested hybrid model has been verified on the benchmark of multiple data and the outcome of the suggested model depicts the enhanced correctness in the detection of varied malware in real-time.

Performance Evaluation:

The performance evaluation of our proposed hybrid model based on the decision tree and the KNN techniques was done by utilizing the five diverse metrics which are F1 amount, Precision, as well categorization correctness ACC, recall, and time consumption, respectively. Each of the matrices is to be generally utilized in machine learning. The first metric is precision may be represented as given in the following equation 1.

$$(Precision) = \frac{TP\ (True\ Positive)}{TP(True\ Positive)+FP'(False\ Positive)} \quad (1)$$

Herein TP represents the overall quantity of the malware sampling that is accurately identified. Further, FP represents the number of benign apps that are incorrectly identified in place of the Android malware. Further, the term precision indicates the ratio of detected malware sampling to original malware sampling. The term Recall may be defined in the following equation 2.

$$(Recall) = \frac{TP\ (True\ Positive)}{TP(True\ Positive)+FN'(False\ Negative)} \quad (2)$$

In above equation 2, the term FN represents the overall quantity of android malicious sampling which are not identified. Herein, the term recall imitates the proportion of malware sampling recognized in real-time malware sampling. The term ACC may define in the following equation 3.

$$(ACC) = \frac{TP\ (True\ Positive)+TN\ (True\ Negative)}{TP(True\ Positive)+TN(True\ Negative)+FP(False\ Positive)+FN'(False\ Negative)} \quad \ldots \quad (3)$$

Herein, the term TN indicates quantity of the benign apps which are accurately categorized. Further ACC is a complete assessment of the classifier and represents the promotion of the entire amount of the apps that are accurately categorized whether in place of benign or malware. The high value of the ACC indicates the high performance of the proposed model based on the decision tree and the KNN technique. The term F1 may be illustrated in the following equation 4. The term F1 may be defined as harmonic mean in betwixt Recall as well as Precision.

$$(F1) = \frac{2\times Precision\times Recall}{Precision+Recall} \quad (4)$$

4.1 Experiments Procedure:

In this experiment procedure, the authors separated sampling datasets in the training datasets along with the testing dataset in a ratio of 6:2. Further, another dataset that is pseudo testing dataset has been utilized for updating the procedure of decision nodes as well as further for obtaining the decision tree based on the combined techniques that are decision tree along with the KNN technique. At last, entire testing datasets were employed for the evaluation of the performance constraints of the entire classifier. In this trial procedure, we opted for 815 normal samples, and 3268 for the malware sampling datasets. For the decision tree and KNN-based proposed model, the 60.00% training datasets along with the 20% pseudo testing datasets were employed while a lasting 20.00% has been utilized for evaluation of the classifier performance constraints.

Results Analysis:

Figure 3 illustrates the performance measures accuracy of various parameters such as precision, ACC, Recall, and F1 along with the time consumption for the proposed hybrid model. Trial 1 was carried out in three diverse segments the decision tree technique, the KNN technique, and the proposed hybrid scheme of combining techniques of decision tree and the KNN. The precision and ACC, along with the Recall, F1 and time consumption were measured for all three segments 0.91, 0.95, 0.98 and 0.88, 0.89, 0.98 along with 0.91, 0.92, 0.98 as well as 0.91, 0.94, 0.97 and 18.55s, 101.55s, 8s, respectively. Figure 4 illustrates the performance measures accuracy of various parameters such as precision, ACC, Recall, F1 along with the time consumption for the proposed hybrid model along with the CNN method. The precision and ACC, along with the Recall, F1, and time consumption were measured for CNN (Convolutional Neural Network) and the proposed hybrid model, 0.93, 0.98, and 0.96, 0.98 along with the 0.6, 0.98, and 0.96, 0.97 as well as 385.24s and 8s.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5899
***Article Received**: 12 October 2022 **Revised**: 12 November 2022 **Accepted**: 12 December 2022*

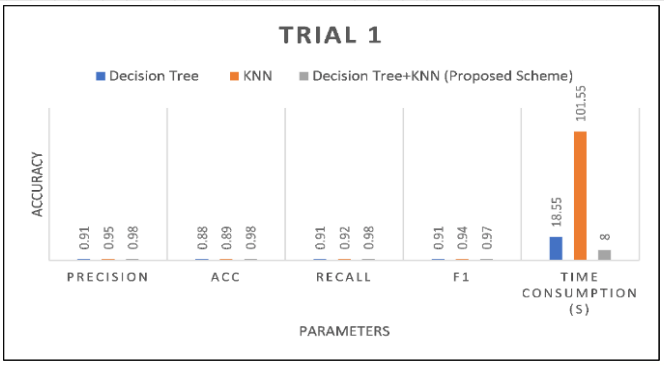

Fig 3: Illustrates the performance measures accuracy of various parameters such as precision, ACC, Recall, F1 along with the time consumption for the proposed hybrid model.

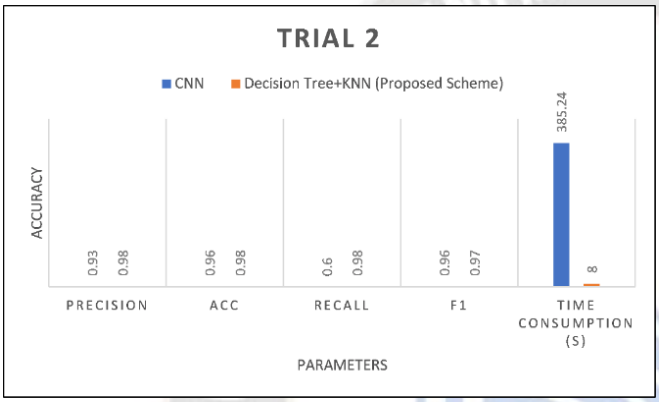

Fig 4: Illustrates the performance measures accuracy of various parameters such as precision, ACC, Recall, F1 along with the time consumption for the proposed hybrid model along with the CNN method.

Figure 5 illustrates the performance measures accuracy of various parameters such as precision, ACC, Recall, and F1 based on different sample sizes is 815 malware samples size and 3268 normal sample sizes. In this experiment, the authors selected two diverse sets of samples size for the performance evaluation of the proposed hybrid model based on the decision tree and the KNN technique. During the experiment, 815 sample size was selected for the normal samples and the 3268-sample size was selected for the malicious samples. Our proposed hybrid model provides pragmatic values of the parameters namely precision, ACC along with the Recall, and F1 such as 0.93, 0.98, 0.96, 0.99 along with 0.94, 0.99, and 0.93, 0.99, respectively. We compared our hybrid model performance parameters which are based on the decision tree and KNN technique along with the existing work done by H. sun et al. [33].

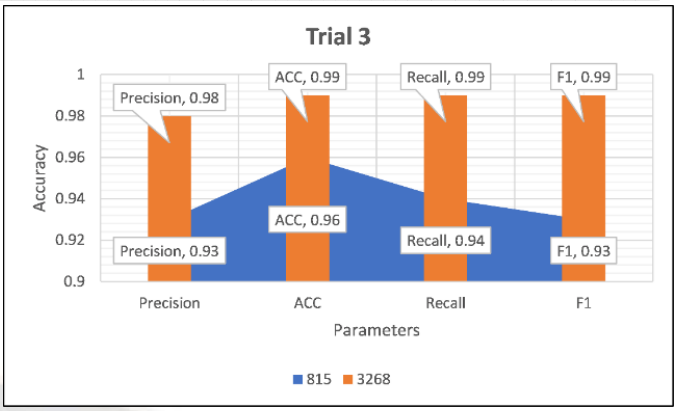

Fig. 5: Illustrates the performance measures accuracy of various parameters such as precision, ACC, Recall, and F1 based on a different sample size that is 815 normal samples size and 3268 malicious sample sizes.

V. Conclusion

The malware in the android based operating systems is becoming a major problem all around the world. There have been developed numerous ways of resolving these issues effectively during the last decade. However, due to the constant technological advancement in the android-based operating systems, diverse kinds of malware such as spyware, viruses or trojans are becoming more harmful and demand more attention towards resolving this problem more pragmatically to provide the required secrecy against the various types of the malware. In this research, the author developed a novel and effective hybrid model that is rooted in the decision tree and the KNN technique. The outcome of the proposed model is more pragmatic and recorded parameters such as precision, ACC along with the Recall, and F1 such as 0.93, 0.98, 0.96, 0.99 along with 0.94, 0.99 and 0.93, 0.99, for the 815 normal sample size and 3268 malicious sample sizes, respectively. This suggested hybrid model is capable of combining the decision tree and the KNN technique. Under the premise of preserving a higher accurate decision pathway, the KNN is being utilized for efficiently minimizing the overfitting threats in the decision tree, and therefore enhances generalization capability. Further, it preserves the eminence of KNN for small as well as large sampling training datasets. The suggested model is verified with a high degree of precision and the outcome of the model depicts that our model offers high speed and consumes less time in comparison to the previously developed models for the android malware detection. In the future, there are vital possibilities for further research to find more pragmatic schemes for android malware detection in less period and provide additional protection against the various category of malware.

Acknowledgement

Authors acknowledge the immense help received from the scholars whose articles are cited and included in references to

this manuscript. The authors are also grateful to authors/ editors / publishers of all those articles, journals and books from where the literature for this article has been reviewed and discussed.

REFERENCES

[1] H. Zhou, X. Yang, H. Pan, and W. Guo, "An Android Malware Detection Approach Based on SIMGRU," IEEE Access, 2020, doi: 10.1109/ACCESS.2020.3007571.

[2] O. C. Abikoye, B. A. Gyunka, and O. N. Akande, "Android malware detection through machine learning techniques: A review," Int. J. online Biomed. Eng., 2020, doi: 10.3991/ijoe.v16i02.11549.

[3] P. Feng, J. Ma, C. Sun, X. Xu, and Y. Ma, "A novel dynamic android malware detection system with ensemble learning," IEEE Access, 2018, doi: 10.1109/ACCESS.2018.2844349.

[4] Y. C. Chen, H. Y. Chen, T. Takahashi, B. Sun, and T. N. Lin, "Impact of Code Deobfuscation and Feature Interaction in Android Malware Detection," IEEE Access, 2021, doi: 10.1109/ACCESS.2021.3110408.

[5] A. T. Kabakus, "What static analysis can utmost offer for android malware detection," Inf. Technol. Control, 2019, doi: 10.5755/j01.itc.48.2.21457.

[6] Z. Ren, H. Wu, Q. Ning, I. Hussain, and B. Chen, "End-to-end malware detection for android IoT devices using deep learning," Ad Hoc Networks, 2020, doi: 10.1016/j.adhoc.2020.102098.

[7] S. Y. Yerima and S. Sezer, "DroidFusion: A Novel Multilevel Classifier Fusion Approach for Android Malware Detection," IEEE Trans. Cybern., 2019, doi: 10.1109/TCYB.2017.2777960Y.

[8] J. Xu, Y. Li, R. H. Deng, and K. Xu, "SDAC: A Slow-Aging Solution for Android Malware Detection Using Semantic Distance Based API Clustering," IEEE Trans. Dependable Secur. Comput., 2022, doi: 10.1109/TDSC.2020.3005088.

[9] T. Kim, B. Kang, M. Rho, S. Sezer, and E. G. Im, "A multimodal deep learning method for android malware detection using various features," IEEE Trans. Inf. Forensics Secur., 2019, doi: 10.1109/TIFS.2018.2866319.

[10] X. Xiao, S. Zhang, F. Mercaldo, G. Hu, and A. K. Sangaiah, "Android malware detection based on system call sequences and LSTM," Multimed. Tools Appl., 2019, doi: 10.1007/s11042-017-5104-0.

[11] X. Liu, X. Du, X. Zhang, Q. Zhu, H. Wang, and M. Guizani, "Adversarial samples on android malware detection systems for IoT systems," Sensors (Switzerland), 2019, doi: 10.3390/s19040974.

[12] J. Lee, H. Jang, S. Ha, and Y. Yoon, "Android malware detection using machine learning with feature selection based on the genetic algorithm," Mathematics, 2021, doi: 10.3390/math9212813.

[13] Y. Yang, X. Du, Z. Yang, and X. Liu, "Android malware detection based on structural features of the function call graph," Electron., 2021, doi: 10.3390/electronics10020186.

[14] X. Chen et al., "Android HIV: A Study of Repackaging Malware for Evading Machine-Learning Detection," IEEE Trans. Inf. Forensics Secur., 2020, doi: 10.1109/TIFS.2019.2932228.

[15] X. Jiang, B. Mao, J. Guan, and X. Huang, "Android Malware Detection Using Fine-Grained Features," Sci. Program., 2020, doi: 10.1155/2020/5190138.

[16] P. Palumbo, L. Sayfullina, D. Komashinskiy, E. Eirola, and J. Karhunen, "A pragmatic android malware detection procedure," Comput. Secur., 2017, doi: 10.1016/j.cose.2017.07.013.

[17] S. Y. Yerima, S. Sezer, and I. Muttik, "Android malware detection using parallel machine learning classifiers," 2014. doi: 10.1109/NGMAST.2014.23.

[18] E. J. Alqahtani, R. Zagrouba, and A. Almuhaideb, "A survey on android malware detection techniques using machine learning Algorithms," 2019. doi: 10.1109/SDS.2019.8768729.

[19] J. D. Koli, "RanDroid: Android malware detection using random machine learning classifiers," 2018. doi: 10.1109/ICSESP.2018.8376705.

[20] R. Agrawal, V. Shah, S. Chavan, G. Gourshete, and N. Shaikh, "Android Malware Detection Using Machine Learning," 2020. doi: 10.1109/ic-ETITE47903.2020.491.

[21] S. Y. Yerima, M. K. Alzaylaee, A. Shajan, and P. Vinod, "Deep learning techniques for android botnet detection," Electron., 2021, doi: 10.3390/electronics10040519.

[22] "Graph Approach for android malware detection using machine learning techniques," Humanit. Nat. Sci. J., 2021, doi: 10.53796/hnsj21115.

[23] M. Kedziora, P. Gawin, M. Szczepanik, and I. Jozwiak, "ANDROID MALWARE DETECTION USING MACHINE LEARNING AND REVERSE ENGINEERING," 2018. doi: 10.5121/csit.2018.81709.

[24] R. Taheri, R. Javidan, M. Shojafar, Z. Pooranian, A. Miri, and M. Conti, "On defending against label flipping attacks on malware detection systems," Neural Comput. Appl., 2020, doi: 10.1007/s00521-020-04831-9.

[25] T. A. A. Abdullah, W. Ali, and R. Abdulghafor, "Empirical study on intelligent android malware detection based on supervised machine learning," Int. J. Adv. Comput. Sci. Appl., 2020, doi: 10.14569/IJACSA.2020.0110429.

[26] B. A. Gyunka and S. I. Barda, "Anomaly detection of android malware using One-Class K-Nearest Neighbours (OC-KNN)," Niger. J. Technol., 2020, doi: 10.4314/njt.v39i2.25.

[27] Y. Pan, X. Ge, C. Fang, and Y. Fan, "A Systematic Literature Review of Android Malware Detection Using Static Analysis," IEEE Access, 2020, doi: 10.1109/ACCESS.2020.3002842.

[28] C. Li, K. Mills, D. Niu, R. Zhu, H. Zhang, and H. Kinawi, "Android Malware Detection Based on Factorization Machine," IEEE Access, 2019, doi: 10.1109/ACCESS.2019.2958927.

[29] K. Liu, S. Xu, G. Xu, M. Zhang, D. Sun, and H. Liu, "A Review of Android Malware Detection Approaches Based

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5899
***Article Received**: 12 October 2022 **Revised**: 12 November 2022 **Accepted**: 12 December 2022*

on Machine Learning," IEEE Access, 2020, doi: 10.1109/ACCESS.2020.3006143.

[30] S. K. Sasidharan and C. Thomas, "ProDroid — An Android malware detection framework based on profile hidden Markov model," Pervasive Mob. Comput., 2021, doi: 10.1016/j.pmcj.2021.101336.

[31] H. Chen, Z. Li, Q. Jiang, A. Rasool, and L. Chen, "A hierarchical approach for android malware detection using authorization-sensitive features," Electron., 2021, doi: 10.3390/electronics10040432.

[32] W. Zhang, N. Luktarhan, C. Ding, and B. Lu, "Android Malware detection using tcn with bytecode image," Symmetry (Basel)., 2021, doi: 10.3390/sym13071107.

[33] H. Sun, G. Xu, Z. Wu, and R. Quan, "Android Malware Detection Based on Feature Selection and Weight Measurement," Intell. Autom. Soft Comput., vol. 33, pp. 585–600, Jan. 2022, doi: 10.32604/iasc.2022.023874.