iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.81574

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Intelligent Apk and USB Malware Detection Engine

T.Pavithra, P. Saraswathi, A.Udayalakshmi, B.V.Lavanya
Department of Computer Science and Engineering Bapatla Women's Engineering College

Abstract: ***The rapid growth in digital computing technology has resulted in the development of numerous malware propaga-tion vectors for both mobile computing and removable storage devices.Androidapplications havebecomea target formalicious applications, and USB storage devices can transmit infected executables.Traditionalsignature-baseddetectiontechniquesfor malware detection fail to identify polymorphic and zero-day malware attacks.***
The paper presents an intelligent malware detection system that utilizes a Genetic Algorithm-based feature selection method andmachinelearningalgorithmsforefficientmalwaredetection. The paper analyzes APK files for Android applications and USB storage devices. The APK files for Android applications use the Androguard framework for static analysis and permission-based feature extraction from the AndroidManifest.xml file. Structural featuressuchasPEheadersandimportedAPIfunctionsfor files from USB storage devices use static analysis. The paper utilizes a Genetic Algorithm for optimizing the selected features forefficientclassification.ThepaperpresentsanArtificialNeural NetworkandSVMclassifierforefficientmalwaredetection. The paper presents an experimental evaluation for the proposed system using a dataset consisting of Android applications andPE files from USB storage devices. The results show that the proposed system can achieve a classification accuracy of 94.2
Index Terms: Android Malware Detection, USB Malware Detection,MachineLearning,GeneticAlgorithm,StaticAnalysis, Portable Executable (PE), Support Vector Machine, Neural Network, Dual-Modal Security.

I. INTRODUCTION

The widespread availability of smartphones and storage devices has increased the number of routes through which malwarecanpropagate.Androiddevicesarethemostpop-ular target due to the open nature of the Android platform. Additionally, the use of USB drives to transfer information between personal computers and business environments is common.However,thewidespreaduseofthesetoolsprovides an incentive to malware creators to use them to distribute malware. Malicious Android applications contain hidden mal-ware that can provide hackers access to sensitive information orsystemresources.Ontheotherhand,USBdrivescancarry malware programs such as infected executable files or scripts to propagate malware. This can result in data breaches, financialcrimes,privacyviolations,orunauthorizedcontrolof thecompromisedsystems.Signature-basedanti-malwaretools arethemostcommonlyusedtoolstodetectmalware.However, they are ineffective against unknown types of malware. This calls for the development of more sophisticated tools to identify unknown types of malware. Recently,machinelearninghasshownpromiseformalware detection, especially when used in conjunction with static analysis.ForAndroid-basedmalware,forinstance,identifying suspicious behavior involves examining the permissions spec-ified in AndroidManifest.xml. For USB-distributed malware, examining the PE file structure, including the data in the headersectionandAPIsimportedbythefile,isalsoindicative. However,thesheernumberofpermissionsandAPIsresultsin a high-dimensional feature space that may negatively impact the performance of the machine learning model.

To address this problem, the paper proposes a dual-modal malware detection scheme that combines static analysis witha GA-based feature selection method. This method seeks the most discriminating features from the Android permission feature set and the USB executable feature set. This improves the accuracy of the classification model. The refined feature sets are then used for classification by machine learning modelslikeANNandSVMfordiscriminatingbetweennormal and malicious files.

A.PaperContributions

Thisworkmakesthefollowingcontributions:

- DualPlatformMalwareDetectionSystem:Theauthors develop a single, unified framework that can scrutinize both Android APKs as well as executable files carried on USB drives.
- Feature Optimization Using Genetic Algorithms:The Genetic Algorithm is used to sift through the high-dimensionalAPKpermissionsetsaswellasPortableEx-ecutable attributes, identifying the features of the highest interest.

- MachineLearning-BasedMalwareClassification:The authors use both Artificial Neural Networks (ANN) as well as Support Vector Machines (SVM) as a classifier for malware detection.
- Improved Detection Accuracy:The authors demon-strate that the use of GA with ANN improves the detec-tion accuracy over traditional machine learning as wellas signature-based techniques.

II. RELATED WORK

A number of research studies have addressed the increasing threat of malware by applying different analysis techniques. Karim et al. proposed a malware detector for Android basedon machine learning classifiers that utilize permission-based features. Though the proposed scheme showed decent results in terms of detection accuracy, the effectiveness was reduced due to the high-dimensional feature space formed by the raw permissions. This is because the feature selection method was not effectively applied for feature optimization. Research on the spread of malware using portable storage media and executable files has shown that the analysis of Portable Executable features is a promising direction for identifyingmalware.ResearchbyNissimetal.andSantos et al. focused on opcode sequences, active learning methods, and PE structural features for identifying Windows-based malware. It has been shown that machine learning algorithms arecapableofidentifyingmaliciousexecutables.However,the effectivenessofthesemethodsis reducedwhenahighnumber of features are used without feature optimization methods.

In the case of Android-based malware detection, Arp et al. proposed a lightweight static analysis tool called DREBINthat utilizes Support Vector Machine classifiers based on features extracted from Android Package Files. DREBIN has shown high effectiveness in terms of detection accuracy for Android-based malware. Nevertheless, the proposed scheme has a limitation in that it does not consider the spread of malware using portable media like USB drives. In addition, other research has shown that machine learning algorithms basedonAPIcallsandapplicationbehaviorarepromising for identifying Android-based malware. For instance, Arp et al. proposed a feature mining technique based on permissions forAndroid-basedmalwaredetection.Inaddition,Aaferet al. proposed a tool called DroidAPIMiner that utilizes API features for Android-based malware detection. In the case of Windows-basedmalwaredetection,opcodesequencesandex-ecutable file features have been used for identifying malicious patterns. Though the proposed methods showed decent results in terms of detection accuracy, feature mining techniques are computationally expensive.

Across the board, these studies show a delicate balance between the precision of malware detection, system speed,andfeaturecomplexity.Mostexistingapproachestargeteither Android applications or standalone executables as standalone entities.Tobridgethisexistingdivide,aunifiedframework for malware detection is proposed, which combines feature analysis for both Android APKs and executables carried on USBdevices.ThissystemusesaGeneticAlgorithm(GA) for feature selection from high-dimensional feature spaces. This approach targets permission-based features and features derived from executables with a view to enhancing detection precision while reducing system complexity.

III. SYSTEM ARCHITECTURE

TheDual-ModalAPKandUSBMalwareDetectionSystem is realized as depicted in the architecture diagram provided. The system attempts to address malware detection from two angles: Android APKs and USB-based executables.

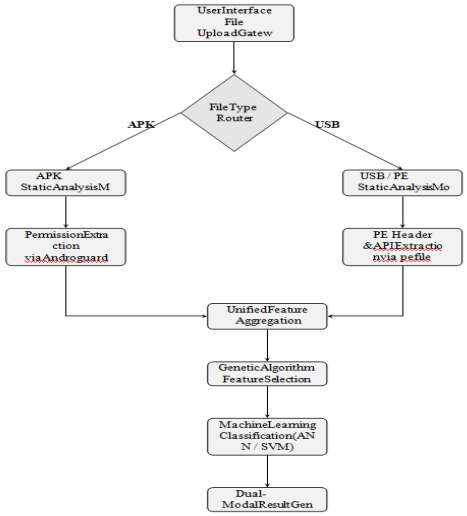


Fig. 1.Architecture of the proposed dual-modal APK and USB malwaredetection system

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

A user initiates this process by uploading a file through the interface provided by the system. The system then identifies thetypeoffileuploadedandprocessesitaccordingly.In the case of Android APKs, it extracts permission featuresfrom AndroidManifest.xml using the Androguard framework. InthecaseofUSB-basedexecutables,itparsesthroughthe PortableExecutablestructuresandAPIfunctionsimported by the executable. The extracted features are then processed through a Genetic Algorithm (GA) that filters out the most relevant features and reduces dimensionality. The optimized featuresetisthenpassedthroughmachinelearningalgorithms suchasArtificialNeuralNetworks(ANN)andSupportVector Machines (SVM) to classify whether the file is malicious or benign. The result is then presented back to the user through the interface provided by the system.

A.Workflow:

Theprocessofmalwaredetectionstartswhentheuserdrops thefileintothewebinterface.ThefilecanbeanAndroidAPK or an executable that you would normally carry around on a USB drive.

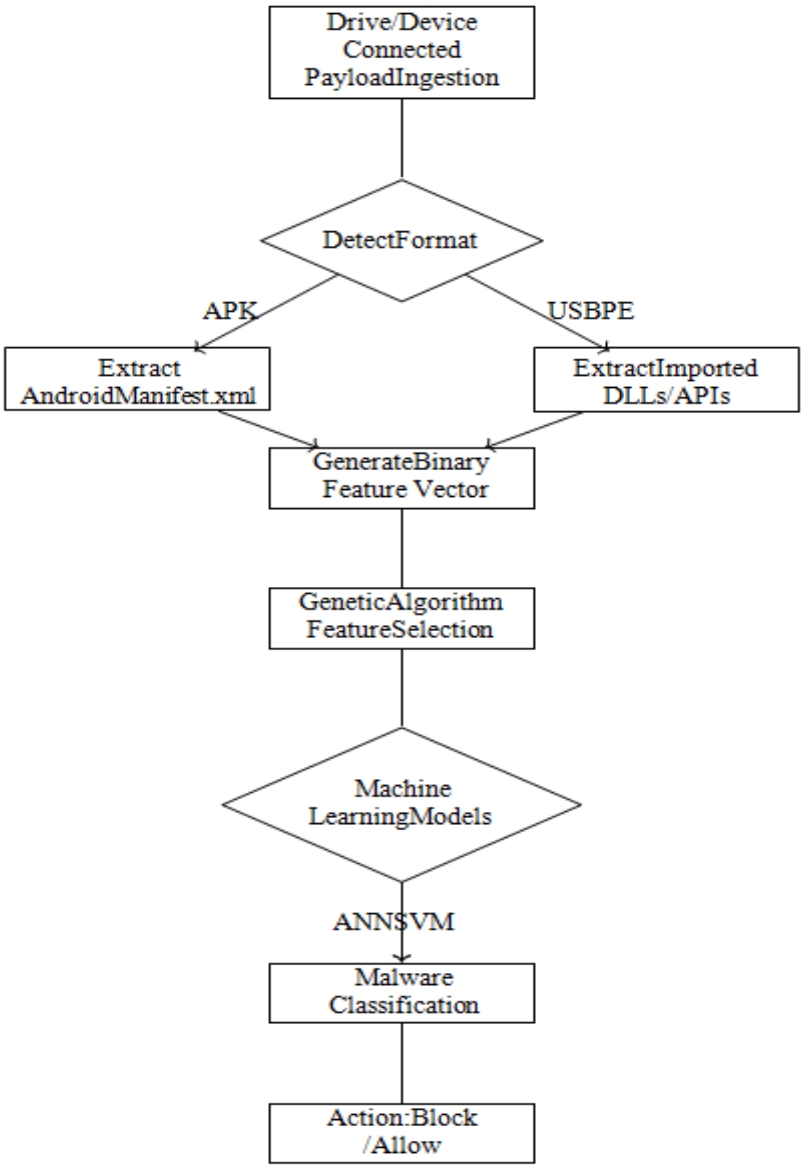


Fig. 2.Unified workflow of the proposed APK and USB malware detectionsystem

The system first determines the type of file, then performsastaticanalysisusingtheappropriatemodule.Themodule for APKs uses Androguard to extract the features of the permissions from the AndroidManifest.xml file. The module forexecutablefilesexaminestheheadersandAPIinformation for Portable Executable (PE). The extracted features are then refined using a Genetic Algorithm (GA) to select the most important features. The refined feature set is then used to train machinelearningclassifierssuchasArtificialNeuralNetworks (ANN) and Support Vector Machines (SVM) to determine whether the file is malicious or not. The results are then presented to the user via the interface.

IV. METHODOLOGY

A. *PayloadAnalysisEngine:*

The malware detection mechanism analyzes the following data streams: Android apps in APK format and Windows executables in PE format that are carried by a USB drive. The permissionsintheAndroidapp'sAndroidManifest.xmlfileare obtained by using the Androguard library. The permissions indicate the capabilities that the app expects to use. The PEfile carried by the USB drive is parsed by using a library such as pefile. The file is examined for section headers, packed resources, and native API calls such as CreateRemoteThread, VirtualAllocEx, and LoadLibraryA. The presence of these characteristicsisastrongindication ofmaliciouscodethat is commonly found in malware.

B. *Multi-VectorFeatureExtraction:*

TomakethedetectionprocesseasierforbothAndroid and Windows executables, a single list of features that en-compasses both Android permissions and Windows API calls is maintained. The list of features for a given file type is converted to a binary format for analysis. The unified feature vector for a file type is defined as follows:

LetXbeafeaturevector:

$$X=\{x_1,x_2,\ldots,x_n\}$$

whereeachelementinthevectorisdefinedas:

$$x_i=\begin{cases}1, & \text{if permission } i \text{ is requested}\\ 0, & \text{otherwise}\end{cases}$$

C. *GeneticAlgorithmforFeatureSelection:*

With the large number of permissions and API features available, the feature space is huge. To deal with the com-plexity of the feature space, a Genetic Algorithm is utilized to selectthemostimportantfeaturesformalwareclassification.A random set of candidate feature subsets is first generated. The fitness of each subset is determined based on the ability of a machine learning model to classify the data with the features. The GA iteratively evolves the feature subsets with crossover and mutation operations. The GA converges to an optimized feature subset that reduces the dimensionality of the feature spaceconsiderablywhileretainingthemostimportantfeatures that indicate malware presence.

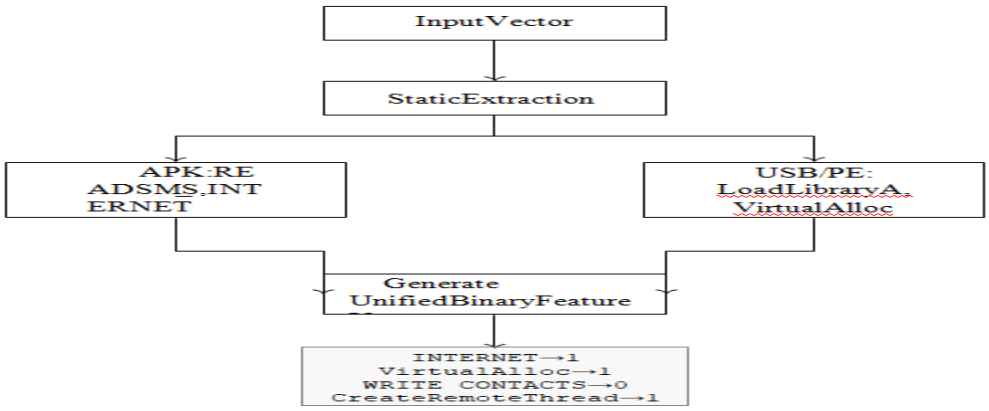


Fig. 3.Dual-modalfeatureextractionprocess

D. MachineLearningModels:

The classifier is then trained using the improved feature vectors created by the Genetic Algorithm.

- ArtificialNeuralNetwork(ANN): The ANN is uti-lized to create a model that recognizes the complex relationships between the permissions and API calls. The complex relationships are nonlinear. The ANN model is abletolearnthepatternsofmalwareforboththeAndroid and Windows platforms.
- Support Vector Machine (SVM): The SVM is utilized as an alternative model. The optimized features are mappedtoahigh-dimensionalspace.TheSVMidentifies the best hyperplane to divide the malware from the legitimate applications and executables.

V. IMPLEMENTATION

The proposed dual-modal malware detection system was implemented using the Python programming language (ver-sions 3.8 to 3.11). The system utilizes the Androguard library to carry out a static analysis of Android APK files to obtain permission-basedfeatures.FortheanalysisofUSB-propagated executablefiles,thesystemutilizesthepefilelibrarytoanalyze Windows Portable Executable (PE) file structures to obtain relevant API imports and structural attributes.

The machine learning model was implemented using Ten-sorFlow (Keras) for developing the Artificial Neural Network (ANN) model and Scikit-learn for developing the Support VectorMachine(SVM)classifier.Moreover,acustomGenetic Algorithm (GA) was implemented to optimize the feature set obtained from combining permissions and API features to obtain relevant features for classification purposes.

The system was tested using a hybrid dataset comprisingthe android dataset-v2 dataset and a dataset comprising USB-propagatedPEmalwaresamples.Thisallowedthesystemto learn from both Android applications and Windows executa-bles, ensuring that it could detect malware from all attack vectors.

VI. EXPERIMENTAL RESULTS

The malware detection approach has been tested on amixed dataset that includes both Android apps and Windows executable files, which are normally propagated through USB storagemedia.Thedatasetincludesover3,000samples,which are a mix of clean apps and malware samples from various malware families. The dataset is well-balanced to include an equal number of good and bad samples. Figure 4 shows a pie chart representing the dataset for evaluation.

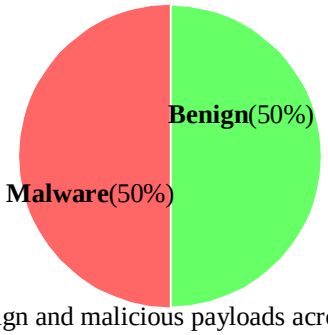


Fig. 4.Distribution of benign and malicious payloads across APK and USBexecutable datasets.

A. PerformanceMetrics:

To evaluate the proposed malware detection system, the usual metrics for assessing the performance of a machine learningmodelareused.Thesemetricsareaccuracy,precision, recall, and F1-score.

Theaccuracyofthesystemiscalculatedusing:

$$Accuracy=\frac{TP_{apk}+TP_{usb}+TN_{apk}+TN_{usb}}{\text{Total Samples}}$$

where *TP*represents True Positives and *TN*represents True Negatives for APK and USB samples.

B. ModelPerformanceComparison:

The performance of both models, SVM and ANN, has been comparedbasedontheaccuracyachievedbybothmodels.The table I shows the performance comparison of both models.

TABLEI
PERFORMANCECOMPARISONOFMACHINELEARNINGMODELS

Model	Accuracy	Precision	Recall	F1-Score
SVM	92.4%	90.8%	91.5%	91.1%
ANN	94.2%	93.5%	92.8%	93.1%

The results indicate that the Artificial Neural Network (ANN) provided the best performance overall, as it performed better than the SVM model in all the evaluation criteria.

C. ConfusionMatrixAnalysis:

Togodeeperintothelevelofperformance,wedeveloped a confusion matrix for our unified ANN-based detection.

TABLEII
CONFUSIONMATRIXFORTHEUNIFIEDANNCLASSIFIER

	PredictedMalware	**PredictedBenign**
ActualMalware	**250(125APK,125USB)**	**12**
ActualBenign	**8**	**270**

Reviewing the confusion matrix, the dual-modal detection approach clearly excels at sorting malicious and benign sam-ples. There is only a small amount of false positives and false negatives. This further supports the belief that the use of the GeneticAlgorithmwithANN-basedclassificationisusefulfor malware detection.

VII. COMPARISON WITH EXISTING METHODS

Inordertodeterminetheefficacyoftheproposeddualmodalmalwaredetector,ithasbeencomparedwithseveralconventionalmethodsdescribedintheliterature.Signaturebasedconventionalmethodsareonlyeffectiveforknownmalware;however,theyaretypicallyaccuratetoonly82%due totheinabilitytorecognizepolymorphicorzero-daymalware. Incomparison, machinelearningtechniqueslikeRandom Forestand SupportVectorMachine(SVM)havebeenmore accurate in detecting malware. In fact, Random Forest has an averageaccuracyof88.5%,whereastheconventionalSVM has an average accuracy of 89.2%. However, these techniques areoftenonlyusedwithsingle-platformdata,whichmay notbeoptimalduetothelackofoptimizedfeatureselectiontechniques.

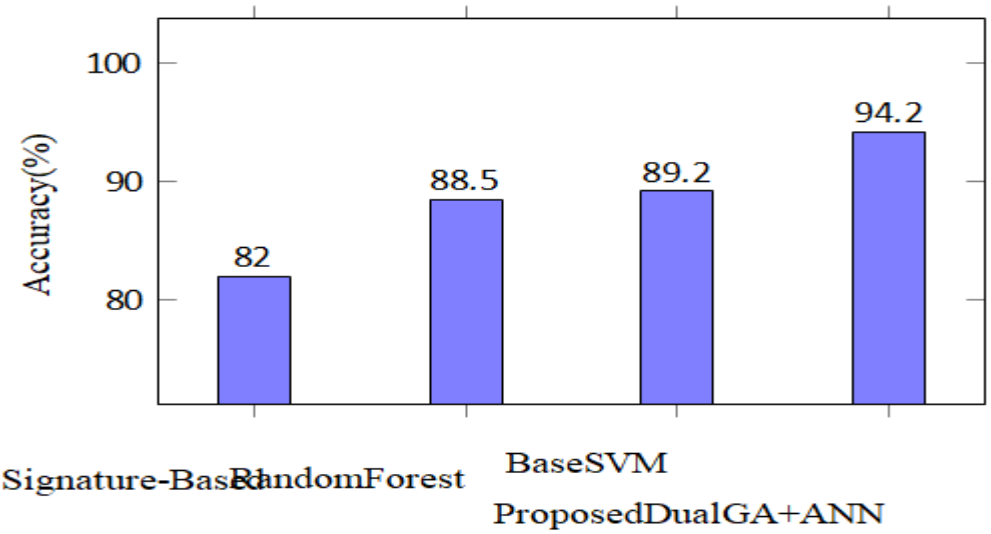


Fig.5.Accuracycomparisonbetweenexistingmethodsandtheproposeddual-modal malware detection system.

Fig.5illustratestheaccuracyimprovementachievedby the proposed GA+ANN model compared with traditional approaches.

In the proposed system, the feature selection process has been optimized using Genetic Algorithm (GA) techniques combinedwithanArtificialNeuralNetwork(ANN)classifier. In addition, the proposed system has been evaluated withboth Android APK files and USB executable files, achieving an overall accuracy of 94.2%, which is higher than the conventional methods.

TABLEIII
COMPARISONWITHEXISTINGMALWAREDETECTIONMETHODS

Method	ThreatScope	Accuracy
Signature-BasedDetection	SingleVector	82.0%
RandomForest	SingleVector	88.5%
BaseSVM	SingleVector	89.2%
ProposedDualGA+ANN	Dual(APK&USB)	94.2%

Table III presents a comparison between the proposed approach and several existing malware detection techniques.

VIII. CONCLUSION

Thecontributionofthecurrentworkistheintroduction of a dual-modal malware detector that is able to identify malware from both Android application and USB-based ex-ecutable file types. This approach combines static analysisand genetic algorithm-based feature selection to reduce the extensive features space and improve the efficiency of the model for classification. To classify the features, the current work relies on machine learning techniques such as Artificial NeuralNetworks(ANN)andSupportVectorMachines(SVM) to interpret the features.

Theresultsoftheexperimentsindicatetheimportanceof optimizing the features for more distinct and efficient malware detection. The results of the experiments indicate the importanceofoptimizingthefeaturesformoredistinctandef-ficientmalwaredetection.Thisisbecausetheproposedmodel, relyingontheANNtechnique,attainsa94.2%accuracylevel, surpassing other traditional malware detection approaches. This indicates the importance of the proposed framework, whichcombinesAPK andUSBfeaturesforeffectivemalware detection.

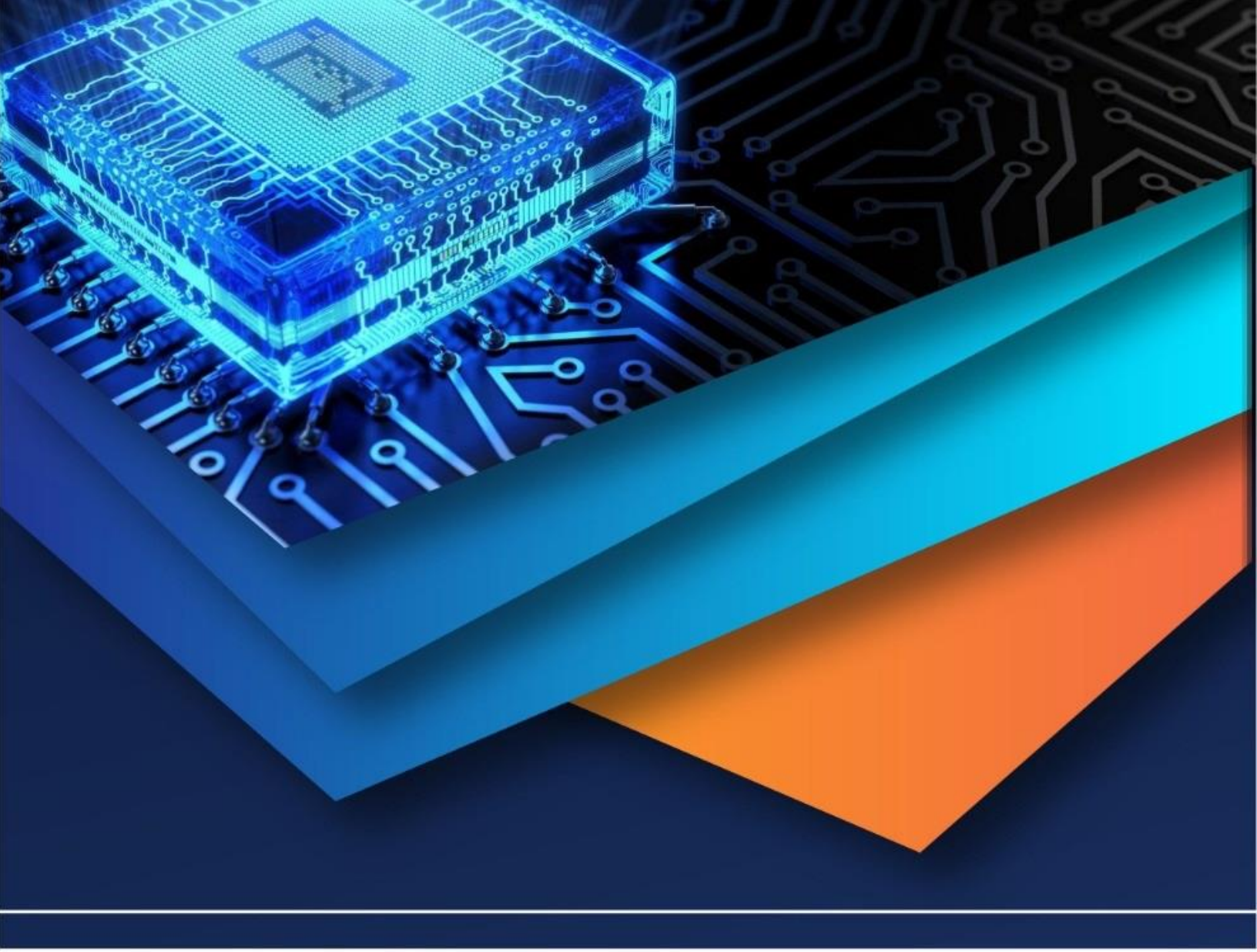
IX. FUTURE WORK

The future direction for research would involve developing the malware detection framework by incorporating more mal-ware analysis techniques and sophisticated machine learning models. This could involve developing a hybrid approach by integrating static feature extraction with dynamic sandbox technology to monitor the behavior of suspicious APK files and executable files in real-time. Another direction for future research would involve incorporating Graph Neural Networks (GNNs) for analyzing the structure of API calls, which would helpindetectinghighlyobfuscatedmalware.Anotherdirection for future research would involve developing a real-time mal-waredetectionsystemthatusesedgeAItechnologytomonitor theconnectionsof USBdevice sand application installations.

REFERENCES

[1] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, and K. Rieck,"DREBIN:EffectiveandExplainableDetectionofAndroidMalwarein Your Pocket," in Proc. Network and Distributed System SecuritySymposium (NDSS), 2014.
[2] S. Arshad et al., "SAMADroid: A Novel 3-Level Hybrid MalwareDetection Model for Android Operating System," IEEE Access, vol. 6,
[3] pp.4321–4339,2018.
[4] Y. Aafer, W. Du, and H. Yin, "DroidAPIMiner: Mining API-LevelFeatures for Robust Malware Detection," in SecureComm, pp. 86–103,2013.
[5] J. Su, V. Rastogi, and S. Hicks, "DroidSieve: Fast and AccurateClassification of Obfuscated Android Malware," in Proc. ACM Conf.Data and Application Security and Privacy (CODASPY), pp. 209–220,2016.
[6] W.Encketal.,"TaintDroid:AnInformation-FlowTrackingSystemfor Realtime Privacy Monitoring on Smartphones," in Proc. USENIXSymposium on Operating Systems Design and Implementation (OSDI),2010.
[7] R. Karim, S. Luo, G. Zheng, and B. Li, "An Evaluation of MachineLearning Algorithms for Android Malware Detection," in Proc. IEEEInt. Conf. Smart Computing (SMARTCOMP), 2017.
[8] Y. Zhou and X. Jiang, "Dissecting Android Malware: Characterizationand Evolution," in Proc. IEEE Symposium on Security and Privacy, pp.95–109, 2012.
[9] M. A. A. Ghorbani, Y. Li, and A. Mahanti, "Optimizing Permission-Based Malware Detection for Android Using Genetic Algorithms," inProc. IEEE Int. Conf. Trust, Security and Privacy in Computing andCommunications (TrustCom), 2019.
[10] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, and Y. Weiss, "An-dromaly: A Behavioral Malware Detection Framework for AndroidDevices," Journal of Intelligent Information Systems, vol. 38, pp. 161–190, 2012.

[11] M.Lindorfer,M.Neugschwandtner,L.Weichselbaum,andY.Fratanto-nio,"ANDRUBIS—1,000,000AppsLater:AViewonCurrentAndroidMalware Behaviors," in Proc. Int. Workshop on Building AnalysisDatasets and Gathering Experience Returns for Security (BADGERS),2014.
[12] N. Nissim, R. Moskovitch, L. Rokach, and Y. Elovici, "Novel ActiveLearning Methods for Enhanced PC Malware Detection in WindowsOS," Expert Systems with Applications, vol. 41, no. 13, pp. 5843–5854,2014.
[13] I. Santos, F. Brezo, X. Ugarte-Pedrero, and P. G. Bringas, "OpcodeSequences as Representation of Executables for Data-Mining-BasedUnknown Malware Detection," Information Sciences, vol. 231, pp. 64–82, 2013.
[14] N. Peiravian and X. Zhu, "Machine Learning for Android MalwareDetection Using Permission and API Calls," in Proc. IEEE Int. Conf.Tools with Artificial Intelligence (ICTAI), pp. 300–305, 2013.
[15] C. Yang, Z. Xu, G. Gu, V. Yegneswaran, and P. Porras, "DroidMiner:Automated Mining and Characterization of Fine-Grained MaliciousBehaviors in Android Applications," in Proc. European Symposium onResearch in Computer Security (ESORICS), 2014.
[16] Z.AungandW.Zaw,"Permission-BasedAndroidMalwareDetection,"International Journal of Scientific & Technology Research, vol. 2, no.3, pp. 228–234, 2013.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY