ISSN: 2321-2152 **IJJMECE** International Journal of modern electronics and communication engineering

E-Mail

editor.ijmece@gmail.com editor@ijmece.com

www.ijmece.com



Android malware detection using Genetic Alogrithm based Optimized Feature Selection And Machine Learning

¹K.Rajesh, ²K.Ashrith reddy, ³K.Uday kiran, ⁴K.Naveen kumar, ⁵E.Parushu Ramu,

^{1,2,3,4} U.G.Scholor, Department of ECE, Sri Indu College Of Engineering & Technology, Ibrahimpatnam, Hyderabad.

⁵Associate Professor, Department of ECE, Sri Indu College Of Engineering & Technology, Ibrahimpatnam, Hyderabad.

ABSTRACT

Android platform due to open source characteristic and Google backing has the largest global market share. Being the world's most popular operating system, it has drawn the attention of cyber criminals operating particularly through wide distribution of malicious applications. This paper proposes an effectual machine-learning based approach forAndroidMalwareDetectionmakinguseof

evolutionary Genetic algorithm for discriminatory feature selection. Selected features from Genetic algorithm are used to train machine learning classifiers and their capability in identification of Malware before and after feature selection is compared. The experimentation results validate that Genetic givesmostoptimizedfeaturesubset algorithm helping in reduction of feature dimension to less than half of the original feature-set. Classification accuracy of more than 94% is maintained post feature selection for the machine learning based classifiers, while working on much reduced feature dimension, thereby, having a positive impact on computational complexity learning of classifiers.

INTRODUCTION

Android Apps are freely available on Google Playstore, the official Android app store as well as third-party app stores for users to download. Due to its open source nature and popularity, malware writers are increasingly focusingondevelopingmaliciousapplications for Android operating system. In spite of various attempts by Google Playstore to protect against malicious apps, they still find their way to mass market and cause harm to users by misusing personal informationrelated to their phone book, mail accounts, GPS location information and others for misuse by third parties or else take control of the phones remotely. Therefore, there is need to perform malware analysis or reverse- engineering of such malicious applications which pose serious threat to Android platforms. Broadly speaking, Android Malware analysis is of two types: Static Analysis and Dynamic Analysis. Static analysis basically involves analyzing the code structure without executing it while dynamic analysis is examination of the runtime behavior of Android Apps in constrained environment. Given in to the ever-increasing

variantsofAndroidMalwareposingzero-day



ISSN 2321-2152 <u>www.ijmece.com</u> Vol 13, Issue 2, 2025

threats, an efficient mechanism for detection of Android malwares is required. In contrastto signature-based approach which requires regular update of signature database.

Motivation:

In this paper author is using two machine learning algorithms such as SVM (Support Vector Machine) and NN (Neural Networks). App will contains more than 100 features and machine learning will take more time to build model so we need to optimized (reducedataset columns size) features, to optimized features author is using genetic algorithm. Genetic algorithm will choose important features from dataset to train model and remove unimportant features. Due to this process dataset size will reduced be and trainingmodelwillbegeneratedfaster. Inthis paper comparison we are losing some accuracy after applying genetic algorithm but we are able to reduce model trainingexecution time.

Objective:

Android is an open source free operating system and it has support from Google to publish android application on its Play Store. Anybody can developed an android app and publish on play store free of cost. Thisandroid feature attract cyber-criminals to developed and publish malware app on play store. If anybody install such malware app then it will steal information from phone and transfer to cyber-criminals or can give total phone control to criminal's hand. To protect users from such app in this paper author is using machine learning algorithm to detect malwarefrommobileapp.Todetectmalware

from appweneedto extract all codefrom app using reverse engineering and then check whetherappisdoinganymischievousactivity such as sending SMS or copying contact details without having proper permissions. If suchactivitygivenincodethenwewilldetect that app as malicious app. In a single appthere could be more than 100 permissions (examples of permissions are transact, API call signature, on Service Connected, APIcall signature, bind Service, API call signature, attach Interface, API call signature, Service Connection, API call signature, android. os. Binder, API call signature, SEND SMS, Manifest Permission, Ljava. lang.Class. Get Canonical Name, API call signature etc.) which we need to extract from code and then generate a features dataset, if app has proper permission then we will put value 1 in the features data and if not then we will value 0. Based on those features dataset app will be mark as malware or good ware.

LITERATURE SURVEY

Android Malware Detection UsingMachine Learning on Image Patterns

Inthispaper, amalware classification model hasbeenproposedfordetectingmalware samplesintheAndroidenvironment.The proposedmodelisbasedonconvertingsome files from the source of the Android applicationsintograyscaleimages.Some image-based local features and global features, including four different types of localfeaturesandthreedifferenttypesof globalfeatures, have been extracted from the constructed grayscale image datasets and used fortrainingtheproposedmodel.Tothebest of ourknowledge, this type offeatures is used



forthefirsttimeintheAndroidmalware detectiondomain.Moreover,thebagofvisual wordsalgorithmhasbeenusedtoconstruct onefeaturevectorfromthedescriptorsofthe localfeatureextractedfromeachimage.The extractedlocalandglobalfeatureshavebeen usedfortrainingmultiplemachinelearning classifiersincludingRandomforest,k-nearest neighbors, DecisionTree, Bagging, AdaBoost andGradientBoost.Theproposedmethod obtainedaveryhighclassificationaccuracy reached98.75% with a typical computational timedoesnotexceed0.018sforeachsample. The results of the proposed model outperformedtheresultsofallcompared state-of-art models in term of both classification accuracy and computational time.

Android mobile security by detecting and classification of malware based on permissions using machine learning algorithms

Android occupies a major share in the mobile application market. Android mobiles have become an easy target for the attackers. The main reason is the user ignorance in the process of installing and usage of the apps. Android malware can be detected based on he permissions it requests from the user. Several machine learning algorithms arebeing used in the detection of android malware based on the list of permissions enabled for each app. This paper makes an attempt to study the performance of some of the machine learning algorithms, viz., naïve Bayes, J48, Random Forest, Multi-class classifier and Multi-layer Google perceptron.

playstore2015and2016appdataareused

for normal apps and standard malware data sets are used in the evaluation. Multi-class classifier was found to be outperforming the other algorithms in terms of classification accuracy. Naïve Bayes classifier has outperformed as far as model construction time is concerned.

An Android Behaviour Based Malware Detection Method using Machine Learning

In this paper, we propose An Android Behavior-Based Malware Detection Method using Machine Learning. We improve an Android application sandbox, Droidbox, by inserting a view-identification automatic trigger program which can click mobile applications in the meaningful order. Taking advantage of Droidbox result, we collect the behavior such as network activities, file read/write and permission as the feature data and use different machine learning algorithms classify malware and evaluate to the performance. We use a large number of malware and normal application samples to prove that our method has high accuracy.

SYSTEMANALYSIS

EXISTINGSYSTEM

The main contribution of the work isreduction of feature dimension to less than half of original feature-set using Genetic Algorithm such that it can be fed as input to machine learning classifiers for training with reduced complexity while maintaining their accuracyinmalwareclassification.Incontrast to exhaustive method of feature selection which requires 2Ndifferent testing for combinations, where Nisthenumber of



ISSN 2321-2152 www.ijmece.com Vol 13, Issue 2, 2025

features, Genetic Algorithm, a heuristic searching approach based on fitness function has been used for feature selection. The optimized feature set obtained using Genetic algorithm is used to train two machine learning algorithms: Support Vector Machine and Neural Network. It is observed that a decent classification accuracy of more than 94% is maintained while working on a much lower feature dimension, thereby, reducingthe training time complexity of classifiers.

PROPOSEDSYSTEM

- Two set of Android Apps or APKs: Malware/Good wareis reverse engineered toextractfeaturessuchaspermissionsand count of App Components such as Activity, Services, Content Providers, etc. These features are used as feature vector with class labels as Malware and Good ware represented by 0 and 1 respectively in CSV format.
- To reduce dimensionality of feature-set, the CSV is fed to Genetic Algorithm to select the most optimized set of features. The optimized set of features obtained is used for training two machine learning classifiers: Support Vector Machine and Neural Network.
- In the proposed methodology, static features are obtained from AndroidManifest.xml which contains all the important information needed by any Android platform about the Apps. Androguard tool has been used for disassemblingoftheAPKsandgettingthe static features.



Fig. 1. Proposed Methodology

Advantagesofproposedsystem:

- Security
- Proposed a novel and efficient algorithm for feature selection to improve overall detection accuracy.
- Machine-learningbasedapproachin combination with static anddynamic analysis can be used to detect new variants of Android Malware posing zero-day threats.

IMPLEMENTATION

MODULES:

Featureselectionisanimportantpartin machine learning to reduce data dimensionality and extensive research carried outforareliablefeatureselectionmethod.For feature selection filter method and wrapper method have been used. In filter method, features are selected on the basis of their scores in various statistical tests that measure



the relevance of features by their correlation

withdependentvariableoroutcomevaria ble.

Wrapper method finds a subset of features by measuring the usefulness of a subset offeature dependent with the variable. Hence filtermethodsareindependentofanymachine learning algorithm whereas in wrappermethod the best feature subset selected depends on the machine learning algorithm used to train the model. In wrapper method a subset evaluator uses all possible subsets and then uses a classification algorithm to convince classifiers from the features in each subset. The classifier considers the subset of feature with which the classificationalgorithm performs the best. To find the subset, the evaluator uses different search techniques like depth first search, random search, breadth first search or hybrid search. The filter method uses an attribute evaluator along with a ranker to rank all the features in the dataset. Here one feature is omitted at a time that has lower ranks and then sees the predictive accuracy of the classification algorithm. Weights or rank put by the ranker algorithms are different than those by the classification algorithm. Wrapper method is useful for machine learning test whereas filter method is suitable for data mining

testbecausedatamininghasthousandsofmillions of features.

- UploadAndroiddataset
- GenerateTrain&testmodel
- Pre-processing
- RunSVM&Neuralnetwork alg

Algorithmsusedinthisproject:-

The steps involved in feature selection using Genetic Algorithm can be summarized as below:

Step 2: Start the algorithm defining an initial set of population generated randomly.

Step 3: Assign a fitness score calculated by the defined fitness function for genetic algorithm.

Step 4: Selection of Parents: Chromosomes with good fitness scores are given preference over others to produce next generation of off-springs.

Step 5: Perform crossover and mutation operations on the selected parents with the given probability of crossover and mutation for generation of off-springs.

Repeat the Steps 3 to 5 iteratively till the convergence is met and fittest chromosome from population, that is, the optimal feature subset is resulted.



SYSTEMDESIGN

SystemArchitecture:



Fig.System Architecture



Results

Andread Makese Detaction	-	٥	Х
Android Malware Detection Using Genetic Algorithm based Optimized Feature Selection and Machine Learning			
Upload Android Madware Dataset			
Generate Train & Test Model Run SYM Algorithm Run SYM Algorithm Run SYM Algorithm Run SYM Algorithm Run Neural Network with Generic Algorithm Accuracy Graph Execution Time Graph Execution Time Graph			
🖬 🔘 Type here to search 🛛 🖟 😭 😭 🔯 👫 🖍 🌒 🖉 🖈 🕼 🖓	\$* 185 1642-	5 2020	Ð

In above screen I am uploading 'AndroidDataset.csv' file and after uploadwill get below screen

Android Malware Detection				- 0
Android Malware Detecti	ion <mark>Using Genetic A</mark>	lgorithm based Optimized	Feature Selection and Machine I	earning.
Upload Android Malware Dataset	E:/manoj/January/A	IndroidMalware/dataset/And	roidDataset.csv	
Generate Train & Test Model Run SVM Alg	orithm Run SVN	I with Genetic Algorithm	Run Neural Network Algorithm	
Run Neural Network with Genetic Algorithm	Accuracy Graph	Execution Time Graph		
hanoj January Android Malware (dataset Android Dataset.csv lo	aded			
				11/1
H O Type here to search	🤾 🖪 🖉 👔	🖺 🔲 🦉 🧏 🛛	ê 🦉 👰 👘	^ ((12 C) (10 C) (10 C)

Inabovescreenclickon'UploadAndroid Malware Dataset' button and upload dataset.

Generate Train (Run Neural Netv Organ	A lenual	ry > AndroidMalware > dataset	A Fund Com		Run Neural Network Algorithm	n
Run Neural Netv Organ			V ICI Dearch dataset	0		<u> </u>
	te • New folder			B. T. 0		
3	30 Objects 🔥	Name	Date modified	Type		
	Desitop	AndroidDataset.csv	10-12-2020 09-44	Microsoft Ercel C.		
88	Documents	Dataset_Information.csv	10-02-2020 09:22	Microsoft Excel C.	-	
+	Downloads					
2	Music					
-	Pictures					
	Intal Disk (C)					
	Local Disk (E)					
	v <				>	
	File nam	e AndroidDataset.csv		v		
			Open	Cancel		
					The second secon	

Now click on 'Generate Train & Test Model' button to split dataset into train and test part. All machine learning algorithms will take80% dataset for training and 20% dataset to test accuracy of trained model. After clicking that button will get train and test model

Folgod indusid Mahran Datas	Extr	anoi/January/A	ndroidMalware'dataset/And	IroidDataset.csv	
Generate Train & Test Model	Run SVM Algorith	Rm SVM	I with Genetic Algorithm	Run Neural Network Algorithm	
Run Neural Network with Gene	tic Algorithm Acc	uracy Graph	Execution Time Graph		
staret Length : 3799 litted Training Length : 3039					
stavet Length : 3799 litted Training Length : 3039 litted Test Length : 760					
staret Length : 3759 limed Training Length : 3039 limed Test Length : 760					
taset Length : 3599 and Training Length : 3039 and Test Length : 760					
travet Length : 3799 General Training Length : 1439 Genel Test Length : 160					
taret Langt: 1399 Ginel Taris Langt: 1809 Ginel Tes Langt: 100					
tore Longs : 1989 Information Longs : 1889 Bird Tee Longs : 180					

In above screen we can see there are total 3799androidapprecordsarethereand



ISSN 2321-2152 <u>www.ijmece.com</u> Vol 13, Issue 2, 2025

application using 3039 records for training and 760 records for testing. Now we haveboth train and test model and now click on 'Run SVM Algorithm' button to generate SVM model on train and test and get its accuracy

Android Malware Detection						- 0	Х
Android Mahwa	re Detection Using G	ienetic Algori	ithm based Optimiz	ed Feature Selection	a and Machine Learnin	ıg	
Upload Android Malware Dataset	E:/manoj(lannary Andro	idMalware/dataset(ladroidDataset.csv			
Generate Train & Test Model Ru	n SVM Algorithm	Ran SVM wit	h Genetic Algorithm	Rm Neural Net	work Algorithm		
Run Neural Network with Genetic Al	gorithm Accuracy	Graph	Execution Time Gra	ik			
Armany-SAV306211533 Rapet: profile real E-core super 0 0.09 1.00 0.59 02 0 0.07 0.05 025 2000000 0.07 0.05 0.09 2000000 0.09 0.09 0.09 20000000 0.09 0.09 0.09 20000000000000000000000000000000000	ı						
🗄 🛛 Type here to search	Q 🔒 🔒	ŝ <u>I</u>	i () I	a 4 🧃	£ ^ £	54 146320	5

In above screen we got 98% accuracy for SVM and now click on 'Run SVM with GeneticAlgorithm'buttonto chooseoptimize features and then run SVM on optimize features to get accuracy

Upload Android Malware Datase	t Es	manoj/January/A	.adroidMalware/dataset/Aad	roidDataset.csv	
Generate Train & Test Model	Run SVM Algorit	m Run SVN	I with Genetic Algorithm	Run Neural Network Algorithm	
Run Neural Network with Geneti	c Algorithm A	curacy Graph	Execution Time Graph		
eport : precision recall tl-score su	pport				
0 0.93 0.97 0.95 492 1 0.95 0.87 0.90 268 accuracy 0.94 760 macro ang 0.94 0.92 0.93 760 righted ang 0.94 0.94 0.93 760 reflection Matrix : [[479 13]					

In above screen SVM with Genetic algorithm got 93% accuracy. Genetic with SVM accuracy is less but its execution time will be less which we can see at the time of comparison graph.

(Note: when u run genetic then 4 empty windows will open u just close all those 4 windows and let main window to run)

2 CJ	Indovsispte	m2;cndee												- 0	х
a syn no n	ionym of t resource +	ype is deprec no.dtwoel[("	ated; in a resource".	future ve no.ubyte.	ersion of numpy, . 1)])	it will be	unders	tood as (type, (1	1)) []	(1,)type".					^
201	nevals	9/12				mán		nex.							
á		(-10080.	188.82]		7,42614381)	[-18888.		[-18888.	132.						
1		-10000.	183.64		5.932149691	-18888.	90.	[-18888.	115.						
2		i-10000.			5.59732079	i-19999	84.	Ì-18889.	118.						
3		-10000.			4.58195513	-18888.	84.	-18888.	163.						
4		-10000.			3.5745769]	-18888.	83.	-18888.	188.						
5		-10000.			3.37194386]	[-18888.	79.	[-18888.	94.						
6		-10000.			2.81112077]	-18888.	78.	[-18888.	98.						
7		-10000.			3.78534388	-18888.		-18888.							
8		-10000.	88.18		4.22227427	-18888.	70.	[-18888.							
9		-10000.			4.04815008]	-18888.	69.	[-18888.	89.						
10		[-10000.	75.36]		4.68885462]	[-18888.	67.	[-18888.	86.						
11		(+10000.	72.6]		4.43178396]	[-18888.	65.	[-18888.	83.						
12		-10000.	78.32]		4.14458683]	[-18888.	63.	[-18888.	82.						
13		[-10000.	67.66]		3.64751971]	[-18888.	61.	[-18888.	77.						
14		-10000.			3.06104557	[-18888.	60.	[-18888.	76.						
15		-10000.	64.36]		3.59318451]	[-18888.	56.	[-18888.	74.						
16		(-10000.	62.52]		3.87482817]	[-18888.	56.	[-18888.	69.						
17		[-10080.	68.9]		3.67831483]	[-18888.	51.	[-18888.	78.						
18		[-10000.	68.28]		3.97512264]	[-18888.	49.	[-18888.	68.						
19		-10000.	58.64]		4.22355301	[-18888.	49.	[-18888.	78.						
20		-10000.	55.58]		3.41812814]	[-18888.	49.	[-18888.	64.						
21		(-10000.			3.43874336]	[-18888.	48.	[-18888.	6.						
22		-10000.	52.42]		3.73143484	[-18888.	44.	[-18888.	62.						
23		[-10000.	58.18]		3.42747721]	[-18888.	44.	[-18888.	68.						
24		-10000.			4.12318563]	-18888.	42.	[-18889.	8.						
25		-10000.	45.76		3.88619864	-18888.	37.	[-18888.	57.						
26		-10000.	44_46		3.82209367	-18888.	37.	-18888.	- 58.						
27		-10000.	43.16]		3.98577815]	-18888.	36.	[-18888.	55.						
28		-10000.	41.7]		4.25323486]	-18888.	34.	[-18888.	54.						
29		-10998	48.86		4.48186324	-18888.	33.	-18888.	55.						
99 		-10000.	38.66		4.9388563	-18888.	32.	-18888.	54.						
<u>51</u>		-18888.	37.44		4.521/6957	-18888.	12.	-18888.	51.						
82		-18888.	35.16		3.44382193	-18888.	31.	-18888.	45.						
33		-10000.	34.92		3.58248143	-18888.	29.	-18888.	47.						
54 		-18888.	59.26		3.99484557	-18888		-18888.	44.						
55		-18888.	35-8		4.88411557	-18988.	29.	-18888.	48.						
30		-10008.	32.80		216383/982	-10000.	и.	[-18889.	42.						
57 50		-10000.	312.22		3.84164429	-19988.	20.	[-18888.	41.						
30 30		-10000.	32.64		4.355/1541	-10000.	20.	[-18888.	45.						
09 10		-10008.	23.18		4.04821551	-10000.	a.	[-18898.	48.						
47	47	-10998.	<i>a.</i> 9	[e.	3174488855	-10000	ъ.	[-18888.	¥.						v
ł	O Type	here to search		Û	0 🔒	î 🕯		E 🔒 🏮	٢	a	é	°%	∧爰む邻	9494 11-12-2120	5

In above console we can see genetical gorithm chooses 40 features from all dataset features.

Now click on 'Run Neural Network Algorithm' button to test neural network accuracy.

# And	roid Malware Detection					-	σ	×
	Android M:	alware Detection Usin	ng Genetic Al	gorithm based Optimized	Feature Selection and Machine Learnin	g		
	Upload Android Malware Datase	t Et/ma	noj/January/Ai	udroidMalware/dataset/And	iroidDataset.csv			
	Generate Train & Test Model	Run SVM Algorithm	Run SVM	with Genetic Algorithm	Run Neural Network Algorithm			
	Run Neural Network with Geneti	c Algorithm Accu	racy Graph	Execution Time Graph				
ANN	Accuracy : 98.68421052631578							
_								
=	O Type here to search	0 0	1 🙈 🛤	🕮 📄 🎯 📜	e 🕹 🧕 🤞 👌	0 6× 10-0	405 12-2020	0



In above screen neural network also gave 98.64% accuracy. Now click on 'Run Neural Network with Genetic Algorithm' button to get NN accuracy with genetic algorithm

schold Malware Detection						- 0	- >
Android N	lalware Detection Usi	ng Genetic Al	gorithm based Optimized	Feature Selection and Mach	ine Learning		
Upload Android Malware Data:	et E:/ms	noj/January/A	ndroidMalware/dataset/And	roidDataset.csv			
Generate Train & Test Model	Run SVM Algorithm	Run SVM	with Genetic Algorithm	Run Neural Network Algori	thm		
Run Neural Network with Gene	tic Algorithm Accu	racy Graph	Execution Time Graph				
N with Genetic Algorithm Accuracy : 98.03	631585221542						

In above screen NN with genetic got 98.02% accuracy. Now click on 'Accuracy Graph' button to see all algorithms accuracy in graph



In above graph x-axis represents algorithm nameandy-axisrepresentsaccuracyandin

all SVM got high accuracy. Now click on 'Execution Time Graph' button to get execution time of all algorithm



In above graph x-axis represents algorithm name and y-axis represents execution time. From above graph we can conclude that with genetic algorithm machine learningalgorithms taking less time to build model.

CONCLUSION

As the number of threats posed to Android platforms is increasing day to day, spreading mainly through malicious applications or malwares, therefore it is very important to design a framework which can detect such malwares with accurate results. Where signature-based approach fails to detect new variants of malware posing zero-day threats, machine learning based approaches are being used. The proposed methodology attempts to make use of evolutionary Genetic Algorithm togetmostoptimizedfeaturesubsetwhich



can be used to train machine learning algorithms in most efficient way.

FutureEnhancements

From experimentations, it can be seen that a decent classification accuracy of more than 94% is maintained using Support Vector Machine and Neural Network classifiers while working on lowerdimensionfeature-set, thereby reducing the training complexity of the classifiers Further work can be enhanced using larger datasets for improved results and analyzing the effect on other machine learning algorithms when used in conjunction with Genetic Algorithm.

REFERENCES

[1] D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, and K. Rieck, "Drebin: Effective and Explainable Detection of Android Malware in Your Pocket," in Proceedings 2014 Networkand Distributed System Security Symposium, 2014.

[2] N.Milosevic, A.Dehghantanha, and K.K.

R. Choo, "Machine learning aided Android malware classification," Comput.Electr.Eng., vol. 61, pp. 266–274, 2017.

[3] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-An, and H. Ye, "Significant PermissionIdentification for Machine-Learning-Based Android Malware Detection," IEEE Trans. Ind. Informatics,vol.14,no.7,pp.3216–3225, 2018.

[4] A. Saracino, D. Sgandurra, G. Dini, and F. Martinelli, "MADAM: Effective and Efficient Behavior-basedAndroidMalwareDetectionand Prevention,"IEEETrans.DependableSecur.

Comput., vol. 15, no. 1, pp. 83-97, 2018.

[5] S. Arshad, M. A. Shah, A. Wahid, A. Mehmood, H. Song, and H. Yu, "SAMADroid: A Novel 3-Level Hybrid Malware Detection Model for Android Operating System," IEEE Access, vol. 6, pp. 4321–4339, 2018.