Int. J. Pure Appl. Sci. 9(2);381-392 (2023)

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI:10.29132/ijpas.1382344

Detection of Android Based Applications with Traditional Metaheuristic Algorithms

Mehmet Şirin Beştaş[1]*, Özlem Batur Dinler[2]
[1]*Siirt University, Siirt, Türkiye,*
[2]*Siirt University, Faculty of Engineering, Departmant of Computer Engineering, Siirt, Türkiye*
*mehmetsirinbestas@gmail.com, *o.b.dinler@siirt.edu.tr*
Received date:27.10.2023, Accepted date: 08.12.2023

Abstract

The widespread use of devices connected to Android systems in various areas of human life has made it an attractive target for bad actors. In this context, the development of mechanisms that can detect Android malware is among the most effective techniques to protect against various attacks. Feature selection is extremely to reduce the size of the dataset and improve computational efficiency while maintaining the accuracy of the performance model. Therefore, in this study, the five most widely used conventional metaheuristic algorithms for feature selection in the literature, such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Simulated Annealing (SA), Ant Colony Optimization (ACO) and Differential Evolution (DE), was used to select features that best represent benign and malicious applications on Android. The efficiency of these algorithms was evaluated on the Drebin-215 and MalGenome-215 dataset using five different machine learning (ML) method including Decision Tree (DT), K-Nearest Neighbour (KNN), Naive Bayes (NB), Random Forest (RF) and Support Vector Machine (SVM). According to the results obtained from the experiments, DE-based feature selection and RF classifier are found to have better accuracy. According to the findings obtained from the experiments, it was seen that DE-based feature selection and RF method had better accuracy rate.

Keywords: Android, feature selection, malware detection, metaheuristic algorithms, optimization

Geleneksel Metasezgisel Algoritmalar ile Android Tabanlı Uygulamaların Tespiti

Öz

Android sistemlere bağlı cihazların insan yaşamının çeşitli alanlarında yaygın olarak kullanılması, onu kötü aktörler için cazip bir hedef haline getirmiştir. Bu bağlamda, Android zararlı yazılımları tespit edebilen mekanizmaların geliştirilmesi, çeşitli saldırılara karşı korunmada en etkili teknikler arasında yer almaktadır. Özellik seçimi, performans modelinin doğruluğunu korurken veri kümesinin boyutunu azaltmak ve hesaplama verimliliğini artırmak için son derece önemlidir. Bu nedenle, bu çalışmada Genetik Algoritma (GA), Parçacık Sürü Optimizasyonu (PSO), Tavlama Simülasyonu (SA), Karınca Kolonisi Optimizasyonu (ACO) ve Diferansiyel Evrim (DE) gibi literatürde özellik seçimi için en yaygın kullanılan beş klasik meta-sezgisel algoritma, Android'deki iyi huylu ve kötü amaçlı uygulamaları en iyi temsil eden özellikleri seçmek için kullanılmıştır. Bu algoritmaların verimliliği Drebin-215 ve MalGenome-215 veri seti üzerinde Karar Ağacı (DT), K-En Yakın Komşu (KNN), Naive Bayes (NB), Rastgele Orman (RF) ve Destek Vektör Makinesi (SVM) olmak üzere beş farklı makine öğrenme (ML) yöntemi kullanılarak değerlendirildi. Deneylerden elde edilen bulgulara göre, DE tabanlı özellik seçimi ve RF yönteminin daha iyi doğruluk oranına sahip olduğu görülmüştür.

Anahtar Kelimeler: Android, özellik seçimi, zararlı yazılım tespiti, meta-sezgisel algoritmalar, optimizasyon

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI:10.29132/ijpas.1382344

INTRODUCTION

Android is widely regarded as one of the most prevalent operating systems utilised in a variety of smart devices like smartphones, tablets, smartwatches, and other mobile devices (Masum and Shahriar, 2019). Android is widely favoured by a significant portion of global users due to its attributes of being cost-effective, user-friendly, easily accessible, and open source (Islam et al., 2023). As of June 2023, the Android operating system has had d a dominant market share of 70.8% (Statista, 2023). However, despite the global significance of the pervasive use and popularity of the Android platform, the number of cyber security attacks against Android devices has increased rapidly. As a result, the security of Android-based devices is regarded as an essential research topic (Chakravarthy, 2021).

Android malware is a malicious application that accesses sensitive data without the user's knowledge or does actions that the user has not authorised (Albakri et al., 2023). Attackers subject users to a range of attacks pertaining to the domains of reliability, finances, and business processes and operations. These attacks encompass the exposing of personal data and privacy, the theft of identification and credit card information, as well as the manipulation of device functionality such as device locking, displaying advertisements, and unauthorised utilization of device resources (Senanayake et al., 2021). In order to mitigate the risks posed by these attacks, a range of effective security measures have been implemented. These measures include the utilization of firewalls, antivirus software, encryption protocols for safeguarding sensitive information, the implementation of biometric authentication systems for user verification, and the construction of dedicated teams specialising in malware analysis (Tahtacı and Canbay, 2020). On the other hand, the security mechanism of Android, which is reliant on permissions, and the insufficient security scanning (Dağlıoğlu and Doğru, 2019) conducted by Google, persistently contribute to the proliferation of malware (Albakri et al., 2023). Hence, the development of a contemporary, rapid, proficient detection system capable of discerning between legitimate software and malicious apps is of utmost significance.

Feature selection (FS) refers to the task of identifying the most optimal subset from the original feature collection in order to achieve the highest accuracy in prediction (Şahin, 2022; Akinola et al., 2022). Consequently, FS is regarded as an NP-hard optimization problem. Its objective function primarily relies on two elements. minimising the number of features while maintaining maximal classification precision (Kareem, 2022). Since the 1970s numerous feature selection algorithms have been proposed to acquire the most informative subsets that yield superior results. Metaheuristic algorithms are prevalent in the domain of feature selection (Dokeroglu et al., 2022). These algorithms are highly effective and efficient at locating the optimal subset of a dataset. This work primarily emphasises the application of metaheuristic algorithms in addressing feature selection difficulties, owing to their robustness and efficacy (Akinola et al., 2022). Nevertheless, it is important to note that not all algorithms exhibit comparable effectiveness when it comes to tackling various problems. Additionally, it is important to note that every algorithm possesses its own set of constraints (Niyomubyeyi et al., 2020). Therefore, a substantial contribution to the literature can be made by comparing the efficacy of metaheuristic algorithms with statistical analyses.

The study proposes a methodology that centres on wrapper-based feature selection techniques in order to ascertain the significant aspects for the purpose of detecting Android malware. The wrapper techniques involve evaluating the performance of different subsets of features and integrating a metaheuristic algorithm with a classifier. In recent years, many researchers have worked on various metaheuristic algorithms in the field of feature selection to improve performance and efficiency in Android malware detection. However, there is no comparative study that considers the metaheuristic algorithms proposed in this study together. The proposed methodology involved the utilization of metaheuristic algorithms to pick the most appropriate feature set for representing the data related to Android-based malware. Subsequently, machine learning techniques were employed to detect instances of Android malware by using the selected features.

This study provides the following contributions to the literature:

- This paper presents a comparative study of widely used conventional metaheuristic algorithms and ML methods in Android malware detection problems.

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI:10.29132/ijpas.1382344

- The results provide a comprehensive view of how each algorithm performs in detecting Android malware in various scenarios based on two different datasets (Drebin-215 and MalGenome-215) and two different (the dataset is split into 70:30 and 10-fold cross) experimental validation applications.

The article's structure comprises the following sections: The focus of Section 2 is on related works. Section 3 includes the Material and Methods. Section 4 consists of the experimental results and discussion. Section 5 contains the conclusion.

LITERATURE REVIEW

Lee et al. (2021), uses permissions and API calls and tries GA in the feature selection process. The features are subsequently passed to various ML algorithms, including J48, DT, RF, and NB. Using 6000 Android samples in total, the highest achieved accuracy was approximately 97%. Wang et al. (2021), proposed a novel genetic algorithm-based approach for the detection of Android malware. The proposed algorithm is evaluated against the existing genetic algorithm using the UCI dataset. The SV-GA achieves a level of accuracy of 93.6%. Meimandi et al. (2020) demonstrated enhanced performance by integrating the GA and SA techniques with a classification method. Fatima et al. (2019) aimed to validate the performance of SVM and neural networks using a GA approach. Waleed (2019) introduced a method for enhancing SVMs with evolutionary boundary algorithms in order to improve Android malware detection. The optimisation issues were resolved by means of PSO and GA tools which served to improve the performance of the SVM and enhance the accuracy of Android virus detection. Droid-HESVMPSO exhibited a testing accuracy of 96.0%, while Droid-HESVMGA achieved a value of 96.9%. Yildiz and Doğru, (2019) conducted a study in which they examined the Android permission. They employed a feature selection technique based on genetic algorithms and evaluated the performance of this approach using DT, NB, and SVM. Bhattacharya et al. (2019) employed the PSO metaheuristic algorithm as a method for selecting features in order to detect Android malware based on permission analysis. Ling et al. (2019) employed XGboost to detect certain properties, including permissions, intents, APIs, and smali files. They further enhanced the performance of the XGboost model by utilising an ACO. Firdaus et al. (2018) proposed a method for detecting Android malware using genetic selection, which involved the application of a GA.

MATERIAL AND METHODS

The present work employed wrapper based metaheuristics feature selection techniques in order to improve the accuracy of Android malware detection. The Drebin-215 and MalGenome-215 datasets were employed for evaluating the proposed feature selection techniques. PSO, DE, SA, GA, and ACO are used to select important features from the relevant dataset. Then, the obtained feature sets were given as input to DT, KNN, NB, RF and SVM ML methods. For experimental validation, 70:30 (Model-1) and ten-fold cross-validation (Model-2) were applied to the relevant dataset to reveal the performance of the feature selection techniques used in the study. N represented the population size, while T represented the utmost number of iterations. Every algorithm was executed 30 times. Table 1 shows all of the parameters for each algorithm. The experiments were conducted on a Windows 10 Intel(R) Core (TM) i7-8565U CPU 1.80 GHz with 16.0 GB RAM and NVIDIA GeForce MX250 2GB GDDR5. The experiment was conducted using MATLAB R2018a (MathWorks, Natick, MA 01760-2098, USA).

Dataset

All studies were conducted using two datasets, namely Drebin-215 (Arp et al., 2014) and Malgoneme -215 (Zhou and Jiang, 2012). There are 215 features in both datasets. The Drebin-215 dataset has 15,036 samples of apps, of which 9,476 are benign and the other 5,560 are malware. The MalGenome-215 dataset has a total of 3,799 application samples. Among these examples, 2,539 are classified as benign**,** while the remaining 1,260 are classified as malware. This dataset was obtained from the Android malware genome project.

Background of the Conventional Metaheuristic Algorithms

This section provides a concise overview of the conventional metaheuristic algorithms analysed in the present study. During the selection process for these metaheuristics, we prioritised the number of citations, promising results, computational performance, accuracy of prediction, and their primary contributions (Dokeroglu et al., 2022).

ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI:10.29132/ijpas.1382344

Ant Colony Optimization (ACO)

Ant Colony optimization is a metaheuristic approach that draws inspiration from the foraging behaviour of ants in order to address intricate optimization problems (Dorigo et al, 2006). The algorithm employed in this study is rooted on communication-based strategies utilised by ants to establish trail markers and identify optimal pathways.

Differential Evolution (DE)

Differential Evolution is an evolutionary optimization algorithm based on vectors (Storn and Price, 1997). The objective of this algorithm is to identify the optimal solution inside intricate and extensive solution spaces by employing a population-based methodology. DE minimises or maximises the functions that need to be optimised by generating new vectors from vector differences.

Genetic Algorithm (GA)

Genetic Algorithm is an optimization and search algorithm that operates by using the concepts of natural selection and genetic variation (Goldberg and Holland,1988). The objective of this method is to address intricate optimization problems using the emulation of biological evolution mechanisms. The GA employs genetic operators to evolve solution candidates and facilitates the process of identifying the optimal solution.

Particle Swarm Optimization (PSO)

Particle Swarm Optimization is a natural adaptive meta-heuristic algorithm inspired by the movement and cooperation behaviour of bird flocks (Kennedy and Eberhart, 1995). This algorithm aims to solve complex optimization problems by aiming to model the movement of particles and their strategies to find the best solution.

Simulated Annealing (SA)

Simulated Annealing is an optimization process that employs a heuristic approach, drawing inspiration from the thermal equilibrium states observed in physical systems (Van Laarhoven and Aarts,1987). Using energy functions, this algorithm attempts to solve complex optimization problems by approving or rejecting potential solution candidates. SA initially focuses on exploring the solution space with a high acceptance rate at elevated temperatures, but its objective is to ultimately identify superior solutions at lower temperatures.

Table 1. *Algorithm parameter settings*

PSO	DE	SA	GA	ACO

T = 100
N =100
c_1 = 2
c_2 = 2
α = 0.9
	T = 100
	N =100
	cr = 0.9
	f = 0.5
		T = 100
		N = 100
		c = 0.93
		T_0 = 100
			T = 100
			N =100
			cr = 0.8
			mr = 0.01
				T = 100
				N =100
				τ= 1
				eta = 1
				α = 1
				β = 0.1
				rho = 0.2

Background of the Machine Learning Methods

In this section, we provide a concise overview of the prevalent ML techniques employed by wrapper feature selection algorithms.

Decision Tree (DT)

The Decision Trees are a widely employed machine learning algorithm utilised for the purpose of solving classification and regression difficulties. The classification model is constructed using a tree structure via a technique called binary recursive partitioning. It generates a tree consisting of decision nodes and leaf nodes, with decision nodes containing two or more branches and leaf nodes assigning a class or decision (Cihan et al., 2020).

k-Nearest Neighbour (KNN)

The k-Nearest Neighbour approach is a machine learning technique commonly employed in the context of classification and regression prediction problems. This approach is a classification method that relies on assessing the similarity between a new

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI:10.29132/ijpas.1382344

sample to be categorised and previously labelled samples, depending on their distance (Dinler et al., 2021).

Naive Bayes (NB)

The Naive Bayes algorithm is a classification technique that operates on probabilities and is derived from Bayes' theorem. The classification of the test data in this algorithm is defined by its inclusion in the class set that yields the highest value, which is obtained by a sequence of probability calculations performed on the training set (Hailat et al., 2021).

Random Forest (RF)

The Random Forest approach is a commonly employed data mining model for addressing classification and regression challenges. This approach involves conducting training using decision trees that are generated by randomly training numerous distinct subsets. The ensemble of decision trees generated using this approach is commonly referred to as RF. In the context of this classification model, an unidentified test sample is allocated to a certain class based on the decision tree that possesses the highest value. One of the primary benefits associated with the RF model is its ability to effectively address issues related to overfitting and outliers (Ullah et al., 2021).

Support Vector Machine (SVM)

The Support Vector Machine algorithm is widely recognised as a robust and effective machine learning technique employed for the purpose of addressing classification and regression difficulties (Khan et al., 2023). The fundamental premise of this approach is to categorise the samples inside the feature space using a hyperplane. SVM projects features into high-dimensional space in order to identify the hyperplane that provides the most effective separation (Cihan, 2021).

Proposed Model

The proposed methodology for performing the experiments is given in Figure 1. Two datasets of Android applications Drebin-215 and Malgoneme-215 were used for training and testing purposes. To identify significant features, a random subsample of the relevant dataset is provided as input to the wrapper-based feature selection method in this study. In this method, five well-known traditional metaheuristic algorithms are used for feature selection. The selected features are then given as input to five different classification algorithms with two different validation options.

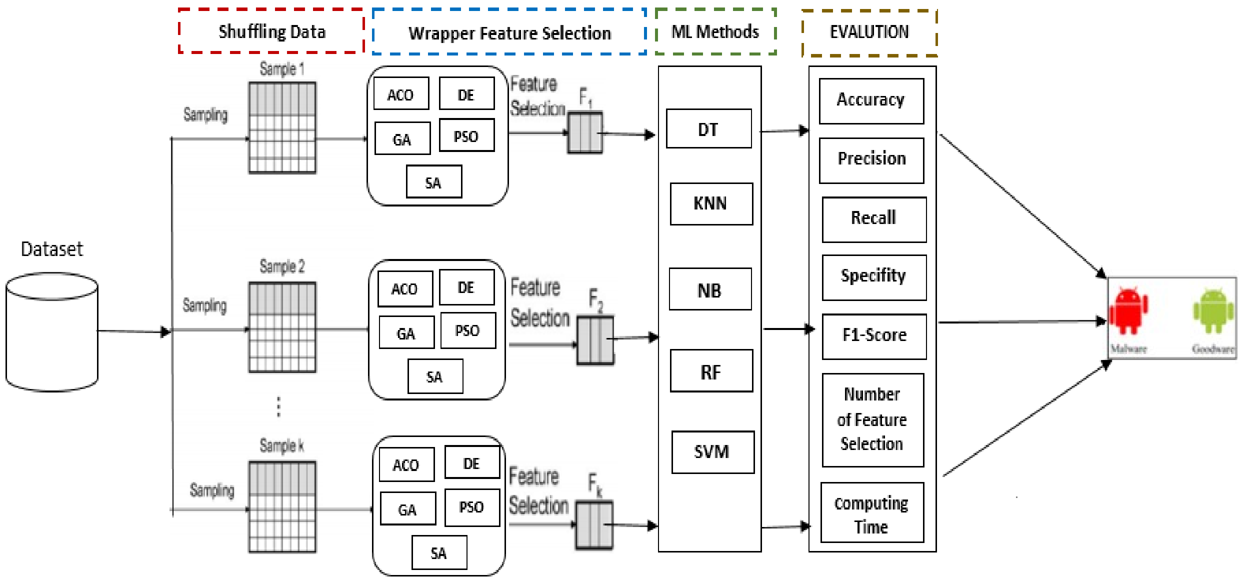

Figure 1. System architecture

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI:10.29132/ijpas.1382344

RESULTS AND DISCUSSION

This section presents and discusses the experimental results.

Performance Measures

A range of evaluation approaches were employed to assess the efficacy of the Conventional Metaheuristic Algorithms. The aforementioned metrics are commonly employed to assess the performance of a classifier, encompassing measures such as accuracy, precision, recall, specificity, and F1-score Equation 1-5.

$$Accuracy\ (Acc) = \frac{TP+TN}{TP+FP+FN+TN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$Specificity = \frac{TN}{TN+FP} \tag{4}$$

$$F1 - Score = \frac{2*Precision*Recall}{Precision*Recall} \tag{5}$$

where *TP*, *TN*, *FP*, and *FN* are true positive, true negative, false positive, and false negative, respectively.

Average Accuracy (AVG_{Acc}): This metric is employed to compute the accuracy of data classification.

Due to the fact that each procedure is repeated 30 times (r_t=30), the average accuracy is computed as following Equation 6:

$$AVG_{Acc} = \frac{1}{r_t}\sum_{k=1}^{r_t} Acc_{Best}^{k} \tag{6}$$

Average Number of Features ($AVG_{|Fs_{Best}|}$): This metric is employed to assess the efficacy of a technique in reducing the quantity of features throughout a certain number of iterations. Its computation is as following Equation 7:

$$AVG_{|Fs_{Best}|} = \frac{1}{r_t}\sum_{k=1}^{r_t} |Fs_{Best}^{k}| \tag{7}$$

Average Computation Time (AVG_{Ct}): This metric is utilised to determine the mean CPU time (s), as depicted in the following Equation 8:

$$AVG_{Ct} = \frac{1}{r_t}\sum_{k=1}^{r_t} Ct_{Best}^{k} \tag{8}$$

Experimental results and performance analysis were conducted using a training set including 70% of the dataset and a test set comprising 30% (Model -1). Additionally, a 10-fold cross-validation strategy (as Model -2) was employed. This study employed five distinct conventional metaheuristic algorithms (namely PSO, DE, SA, GA, and ACO) as feature extraction techniques. The algorithms were assessed using two widely recognised Android malware datasets and five ML classification techniques. The average accuracy, average precision, average recall, average specificity, average F1-Score, average number of selections, and average computational time are displayed in Tables 2 through 8 sequentially. The top results are highlighted in bold.

Table 2 illustrates a comparison of all algorithms' average accuracy results. When comparing several algorithms, it was found that the DE-based feature selection and RF classifier performed the best in two models (Model-1 and Model-2) on the Drebin-215 dataset. Additionally, in Model-2 on the MalGenome-215 dataset, it also earned the highest mean results. The second important indicator is that the average accuracy value of GA-based feature selection and KNN classifier outperforms other algorithms in Model-1 in the MalGenome-215 dataset.

Table 3 presents the mean precision values for all algorithms. The experimental findings demonstrate that the SVM has superiority in both datasets and models.

The average recall values for all algorithms are presented in Table 4. Experimental results show that DE and KNN are superior in Model -1 for Drebin-215 dataset and Model -2 for MalGenome-215 dataset. Moreover, DE and RF are superior in Model-2 for the Drebin-215 dataset and in Model-1 for the MalGenome-215 dataset.

Table 5 presents the mean specificity values. Based on the analysis of the outcomes, it can be inferred that the SVM classifier yielded the most favourable outcomes in both the Drebin-215 and MalGenome-215 datasets, as well as in the two models, namely Model-1 and Model-2, with respect to average specificity. When compared to other outcomes, DE with GA based feature selection in

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI:10.29132/ijpas.1382344

Drebin-215 dataset for Model-1; DE based feature selection in Drebin-215 dataset for Model-1; PSO based feature selection in MalGenome-215 Model -1; and DE with ACO based feature selection in MalGenome-215 for Model -2 attained the highest average specificity.

The mean $F1-Score$ of the comparing all algorithms are presented in Table 6. For $F1-Score$ employed in this investigation, the DE-based feature selection and RF classifier produced the best results in two datasets (Drebin-215 and MalGenome-215) and two models (Model-1 and Model-2).

Table 7 presents the mean value of the chosen features. Based on an analysis of the findings presented in Table 7, it can be inferred that the feature selection technique based on GA demonstrated superior performance in terms of the average number of selected features. This was observed in two datasets, namely Drebin-215 and MalGenome-215, as well as in two models, specifically Model-1 and Model-2, which were utilised in the present study.

Table 8 presents the mean computational duration of the comparative methods. For Model -1 in both datasets, the GA based feature selection and DT classifier is the quickest and requires the least amount of processing effort. The PSO-based feature selection and NB classifier used to the Drebin-215 datasets, as well as the GA-based feature selection and NB classifier employed for the MalGenome-215 dataset, exhibit the shortest computing time and highest speed when compared to alternative approaches for Model-2.

Table 2. *The performance of the average accuracy (%) of all algorithms in Android detection datasets*

	CMA	Drebin-215					MalGenome-215				
		DT	KNN	NB	RF	SVM	DT	KNN	NB	RF	SVM
Model -1	ACO	0.94639	0.97962	0.93337	0.98302	0.9372	0.97021	0.98311	0.95733	0.98657	0.94343
	DE	0.94722	0.98201	0.9324	0.98438	0.93159	0.97457	0.98718	0.96049	0.99064	0.92959
	GA	0.94542	0.98117	0.93098	0.98293	0.93159	0.96986	0.99071	0.95552	0.98721	0.9602
	PSO	0.94576	0.98035	0.93221	0.98168	0.94663	0.97275	0.98581	0.95435	0.98724	0.94267
	SA	0.94024	0.97524	0.97524	0.97962	0.94496	0.97006	0.98171	0.95195	0.98443	0.94496
Model -2	ACO	0.94701	0.98234	0.93337	0.98825	0.94763	0.97396	0.98665	0.95452	0.98874	0.94344
	DE	0.94635	0.984	0.9324	0.9898	0.93744	0.97684	0.98987	0.96067	0.99098	0.94321
	GA	0.94518	0.98379	0.93098	0.98428	0.9515	0.97476	0.98667	0.95498	0.98862	0.96564
	PSO	0.94212	0.98258	0.93221	0.98395	0.95276	0.97477	0.98684	0.95714	0.98873	0.95624
	SA	0.94131	0.97789	0.92639	0.98118	0.95333	0.97158	0.98507	0.95358	0.98763	0.94925

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI:10.29132/ijpas.1382344

Table 3. *The performance of the average precision (%) of all algorithms in Android detection datasets*

	CMA	Drebin-215					MalGenome-215				
		DT	KNN	NB	RF	SVM	DT	KNN	NB	RF	SVM
Model -1	ACO	0.93346	0.97374	0.88875	0.98632	0.99881	0.95316	0.97653	0.92106	0.9881	0.99939
	DE	0.9384	0.97791	0.88494	0.98777	0.99911	0.95737	0.98177	0.92144	0.99182	0.99936
	GA	0.94097	0.97753	0.8974	0.9883	0.99911	0.95705	0.98132	0.93303	0.98667	0.99774
	PSO	0.94017	0.97638	0.89775	0.98677	0.99814	0.95708	0.97951	0.92348	0.98726	0.99989
	SA	0.93782	0.96817	0.88116	0.98291	0.99673	0.95702	0.97265	0.91322	0.98802	0.99673
Model -2	ACO	0.93607	0.97721	0.88875	0.98904	0.99826	0.99154	0.98168	0.9205	0.98921	0.99965
	DE	0.93578	0.97978	0.88494	0.99195	0.99959	0.99469	0.98635	0.92346	0.99322	0.99964
	GA	0.94009	0.98071	0.8974	0.9879	0.99875	0.98771	0.98209	0.93141	0.98788	0.9981
	PSO	0.93365	0.97859	0.89775	0.98799	0.99804	0.98983	0.98191	0.92482	0.98998	0.99916
	SA	0.93716	0.9716	0.88195	0.98475	0.99649	0.96195	0.97852	0.91598	0.98951	0.99944

Table 4. *The performance of the average recall (%) of all algorithms in Android detection datasets*

	CMA	Drebin-215					MalGenome-215				
		DT	KNN	NB	RF	SVM	DT	KNN	NB	RF	SVM
Model -1	ACO	0.92108	0.9711	0.93736	0.96753	0.8312	0.95767	0.97257	0.95406	0.97125	0.83007
	DE	0.91807	0.97336	0.93963	0.96976	0.81577	0.96667	0.97963	0.96323	0.97989	0.78836
	GA	0.90987	0.97144	0.9188	0.96528	0.81577	0.95212	0.97743	0.93307	0.97469	0.8821
	PSO	0.91233	0.9704	0.92197	0.96339	0.85731	0.96129	0.97778	0.94065	0.97416	0.82734
	SA	0.89875	0.96481	0.92144	0.96165	0.854	0.95282	0.97231	0.94577	0.96481	0.854
Model -2	ACO	0.91986	0.97498	0.93736	0.9754	0.85989	0.92942	0.97802	0.94489	0.97672	0.82976
	DE	0.91829	0.9769	0.93963	0.97718	0.83116	0.93516	0.9831	0.96127	0.9795	0.82907
	GA	0.90983	0.97536	0.9188	0.96936	0.86993	0.93556	0.97768	0.93317	0.9777	0.89812
	PSO	0.90874	0.97423	0.92197	0.96836	0.87398	0.93354	0.97837	0.94807	0.97593	0.86881
	SA	0.90255	0.96853	0.92505	0.96404	0.8769	0.95212	0.97643	0.9473	0.97302	0.84746

Table 5. *The performance of the average specifity (%) of all algorithms in Android detection datasets*

	CMA	Drebin-215					MalGenome-215				
		DT	KNN	NB	RF	SVM	DT	KNN	NB	RF	SVM
Model -1	ACO	0.96124	0.98461	0.93102	0.99212	0.99941	0.97643	0.98835	0.95896	0.99417	0.99974
	DE	0.96433	0.98709	0.92816	0.99295	0.99957	0.97849	0.99093	0.95913	0.99597	0.99974
	GA	0.96628	0.98688	0.93812	0.99328	0.99957	0.97867	0.99071	0.96667	0.99343	0.99899
	PSO	0.96538	0.9862	0.93821	0.99241	0.99905	0.97845	0.98979	0.96115	0.99374	0.99996
	SA	0.96459	0.98136	0.92655	0.99017	0.99834	0.97862	0.98638	0.95502	0.99417	0.99834
Model -2	ACO	0.96294	0.98665	0.93102	0.99463	0.99911	0.99606	0.99093	0.93233	0.99471	0.99985
	DE	0.96282	0.98817	0.92816	0.99606	0.9998	0.99752	0.99324	0.96036	0.99668	0.99985
	GA	0.96592	0.98874	0.93812	0.99304	0.99936	0.99421	0.99114	0.9658	0.99404	0.99915
	PSO	0.96171	0.98749	0.93821	0.99309	0.99899	0.99523	0.99104	0.96164	0.99509	0.99963
	SA	0.96404	0.98338	0.92718	0.99124	0.99818	0.98124	0.98935	0.95669	0.99488	0.99976

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI:10.29132/ijpas.1382344

Table 6. *The performance of the average F1-score (%) of all algorithms in Android detection datasets*

	CMA	Drebin-215					MalGenome-215				
		DT	KNN	NB	RF	SVM	DT	KNN	NB	RF	SVM
Model -1	ACO	0.92702	0.97241	0.91234	0.97682	0.90716	0.95526	0.97983	0.937	0.97957	0.90647
	DE	0.92785	0.97562	0.91139	0.97868	0.89812	0.96192	0.98067	0.9418	0.98578	0.88114
	GA	0.92494	0.97446	0.90777	0.97665	0.89812	0.95446	0.97932	0.93297	0.9806	0.93619
	PSO	0.92553	0.97336	0.9096	0.97493	0.92224	0.95906	0.97861	0.93183	0.98064	0.90521
	SA	0.91748	0.96647	0.90047	0.97215	0.91974	0.95478	0.97243	0.92895	0.97624	0.91974
Model -2	ACO	0.92773	0.97609	0.91234	0.98216	0.92381	0.95945	0.97983	0.93233	0.98292	0.90674
	DE	0.92676	0.97834	0.91139	0.98451	0.90761	0.96399	0.98471	0.94191	0.9863	0.90633
	GA	0.92465	0.97802	0.90777	0.97854	0.92987	0.9609	0.97987	0.93218	0.96129	0.94536
	PSO	0.92065	0.9764	0.9096	0.97808	0.93185	0.96084	0.98012	0.93621	0.98289	0.92936
	SA	0.91916	0.97005	0.90287	0.97428	0.93283	0.95695	0.97747	0.93124	0.98119	0.91713

Table 7. *The performance of the average the number of selection of all algorithms in Android detection datasets*

	CMA	Drebin-215					MalGenome-215				
		DT	KNN	NB	RF	SVM	DT	KNN	NB	RF	SVM
Model -1	ACO	141.866	139.5	135.454	134.2	128.8	96.3	92.033	110.833	105.033	102.466
	DE	155.8	156.8	159.454	156.966	156.866	147.1	146.066	150.5	149.066	141.7
	GA	102.6	86.266	89.545	87.545	156.866	45.033	44.166	43.766	43.7	45.133
	PSO	103.5	100.666	98.733	98.5	100.769	80.6	80.433	80.066	81.666	106
	SA	111.411	112.3	107.733	110.133	111.411	107.1	107.3	109.333	107.233	111.411
Model -2	ACO	102.428	139.5	135.454	113.375	122.363	105.6	102.366	104.8	103.866	131.375
	DE	154.437	156.8	159.454	126.25	160.818	145.733	148.166	150.6	146.366	133.5
	GA	84.8571	86.266	89.5455	86.090	86.2667	43.7667	67.625	45.733	45.2333	45.133
	PSO	101.357	100.666	98.733	102.454	100.666	79.733	101	86.633	79.466	79.266
	SA	107.366	112.033	112.3	109.733	108.266	106.233	108.766	108.933	106.933	106.966

Table 8. *The performance of the average computational time (sec) of all algorithms in Android detection datasets*

	CMA	Drebin-215 -215					MalGenome-215 -215				
		DT	KNN	NB	RF	SVM	DT	KNN	NB	RF	SVM
Model -1	ACO	0.13026	0.13659	0.47612	0.65864	1.5592	0.01862	0.03057	0.02187	0.2052	0.38998
	DE	0.1126	0.13026	0.50467	0.67379	1.3738	0.02205	0.02779	0.02614	0.21755	0.32752
	GA	0.04144	0.09715	0.37099	0.6338	1.7092	0.01164	0.02653	0.01635	0.17633	0.3655
	PSO	0.04161	0.107	0.35276	0.62313	2.4899	0.01470	0.02676	0.01848	0.19593	0.42445
	SA	0.0619	1.2369	0.05483	0.57432	4.3565	0.02139	0.05212	0.03296	0.15981	2.5294
Model -2	ACO	0.45712	2.628	0.47612	1.5684	1.5007	7.2624	0.26941	0.15259	1.8842	2.698
	DE	0.88607	1.5989	0.50467	1.5899	1.0723	8.1392	0.31717	0.16622	2.0189	2.6978
	GA	0.36529	1.1871	0.37099	1.5114	1.0109	6.0411	0.16524	0.11991	0.96129	2.2608
	PSO	0.50274	1.6035	0.35276	1.4924	1.5029	6.9175	0.21679	0.13789	1.7848	2.7103
	SA	0.5135	3.1926	0.39199	6.0261	24.5547	0.14551	0.22721	0.15286	1.5579	2.5168

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI:10.29132/ijpas.1382344

Results of the Performance Analysis of the Datasets

According to the experimental results in Table 2, Drebin-215 and Malgoneme-215 datasets have the highest accuracy rate with 98.98% and 99.09% respectively with the combination depending on DE +RF+Model-2. In addition, in Tables 3-8 it is observed that the Drebin-215 and Malgoneme-215 DE+RF+Model-2 technique offers an AVG_{Pr} of 99.19%, AVG_{Rc} of 97.71%, AVG_{Sp} of 99.60%, AVG_F of 98.45%, $AVG_{|Fs_{Best}|}$ of 126.25 and AVG_{Ct} of 1.5899; AVG_{Pr} of 99.32%, AVG_{Rc} of 97.95%, AVG_{Sp} of 99.66%, AVG_F of 98.63%, $AVG_{|Fs_{Best}|}$ 146.3 of and AVG_{Ct} of 2.018, respectively.

The second highest accuracy is the combination of RF+Model-2 with ACO for the Drebin-215 dataset and KNN+Model-1 with GA for the Malgoneme-215 dataset.

Results of the Performance Analysis of the Models

According to the experimental results in Table 2-8, Model-2 has the highest accuracy, specifity, and F1-Score for the two datasets, while Model-1 has the highest precision and specificity for the two datasets. In addition, it is seen that Model-1 generally takes less number of features and less computation time than Model-2.

Results of the Performance Analysis of the ML Methods

Upon analysing the findings of the Drebin-215 and Malgoneme-215 datasets, it is observed that the performance of DT, KNN, NB, RF, and SVM algorithms exhibits both enhancements and declines.

CONCLUSION

This study aims to build a method for detecting Android malware by utilising a comprehensive set of criteria to determine the malicious or benign nature of an Android application. The identification and mitigation of Android malware has emerged as a significant issue, including not just individual users but also business entities and governmental organisations.

This study aims to explore the performance and usefulness of conventional metaheuristic algorithms, which are commonly employed, in addressing the challenge of identifying Android malware. The evaluation of the outcomes was conducted by employing diverse performance metrics for each method, including but not limited to the accuracy, precision, recall, specificity, F1-Score, number of selections, and computational time. It is evident that studies conducted in this particular domain will yield substantial contributions towards addressing challenges across several disciplines. Further studies will investigate the performance and effectiveness of the most widely used recent metaheuristic algorithms in solving the Android malware detection problem.

CONFLICT OF INTEREST

There is no conflict of interest for authorship.

RESEARCH AND PUBLICATION ETHICS STATEMENT

This manuscript does not contain any studies with human participants carried out by any of the authors.

REFERENCES

Akinola, O.O., Ezugwu, A.E., Agushaka, J. O., Zitar, R. A. and Abualigah, L. (2022). Multiclass feature selection with metaheuristic optimization algorithms: a review. Neural Computing and Applications, 34 (22), 19751–19790.

Albakri, A., Alhayan, F., Alturki, N., Ahamed, S. and Shamsudheen, S. (2023). Metaheuristics with deep learning model for cybersecurity and Android malware detection and classification. Applied Sciences, 13 (4), 2172.

Arp D., Spreitzenbarth, M., Hubner, M., Gascon, H., Rieck, K. and Siemens, C.E.R.T. (2014). Drebin: Effective and explainable detection of android malware in your pocket. In Ndss, (14), 23-26. Available from:http://www.deeplearningbook.org. (Accessed on 1 November 2022).

Bhattacharya, A., Goswami, R.T. and Mukherjee, K. (2019). A feature selection technique based on rough set and improvised PSO algorithm (PSORS-FS) for permission based detection of Android malwares. International Journal of Machine Learning and Cybernetics, (10), 1893–1907.

Chakravarthy, S. J. (2021). Wrapper-based metaheuristic optimization algorithms for android malware detection: a correlative analysis of firefly, bat & whale optimization. Journal of Hunan University (Natural Sciences), 48 (10), 928-943.

Cihan, P. (2021). The machine learning approach for predicting the number of intensive car, intubated patients and death: The COVID-19 pandemic in Turkey. Sigma Journal of Engineering and Natural Sciences, (40) 1, 85-94.

Cihan, P., Kalıpsız O. and Gökçe, E. (2020). Computer-aided diagnosis in neonatal lambs. Pamukkale Üniversitesi Mühendislik Dergisi, 26 (2), 385-391.

Dağlıoğlu, A. and Doğru, I.A. (2020). Android işletim sisteminde kötücül yazılım tespit sistemleri. Dicle Üniversitesi Mühendislik Fakültesi, Mühendislik Dergisi, 2 (11), 499-511.

Dinler, Ö.B. and Şahin, C.B. (2021). Prediction of phishing web sites with deep learning using WEKA environment. European Journal of Science and Technology, (24), 35-41.

Dokeroglu, T., Deniz, A. and Kiziloz, H.E. (2022). A comprehensive survey on recent metaheuristics for feature Selection. Neurocomputing, (494), 269-296.

Dorigo, M., Birattari, M. and Stutzle, T. (2006). Ant colony optimization. IEEE Computational Intelligence Magazine, 1 (4), 28-39.

Fatima, A., Maurya R., Dutta M.K., Burget R. and Masek, J. (2019). Android malware detection using genetic algorithm based optimized feature selection and machine learning. In Proceedings of the 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), 220–223.

Firdaus, A., Anuar, N. B., Karim, A. and Razak, M. F. A. (2018). Discovering optimal features using static analysis and a genetic search based method for Android malware detection. Frontiers of Information Technology & Electronic Engineering, 19 (6), 712-736.

Goldberg, D. E. and Holland, J. H. (1988). Machine Learning. Machine Learning, 3 (23), 95-99.

Hailat, M. M., Otair, M. A., Abualigah, L., Houssein, E. H. and Şahin, C.B. (2021). Improving automated arabic essay questions grading based on microsoft word dictionary. Deep learning approaches for spoken and natural language processing.

Islam, R., Sayed, M.I., Saha, S., Hossain, M.J. and Masud, M.A. (2023). Android malware classification using optimum feature selection and ensemble machine learning. Internet of Things and Cyber-Physical Systems, (3), 100-111.

Kareem, S. S., Mostafa, R. R., Hashim, F. A. and El-Bakry, H. M. (2022). An effective feature selection model using hybrid metaheuristic algorithms for iot intrusion detection. Sensors, 22 (4), 1396.

Kennedy, J. and Eberhart, R. (1995). Particle Swarm Optimization. In Proceedings of ICNN'95-International Conference on Neural Networks, IEEE, 1942-1948.

Khan, S.N., Khan S.U., Aznaoui H., Şahin, C.B. and Dinler, Ö.B. (2023). Generalization of linear and non-linear support vector machine in multiple fields: a review. Computer Science and Information Technologies, 3(4), 226-239

Lee, J., Jang, H., Ha, S. and Yoon, Y. (2021). Android malware detection using machine learning with feature selection based on the Genetic algorithm. Mathematics, (9), 2813.

Ling, J., Wang, X. and Sun, Y. (2019). Research of android malware detection based on ACO optimized Xgboost parameters approach. In 3rd International Conference on Mechatronics Engineering and Information Technology (ICMEIT), 364-371.

Masum, M. and Shahriar, H. (2019). Droid-NNet: Deep learning neural network for android malware detection. In 2019 IEEE International Conference on Big Data, IEEE, 5789-5793.

Meimandi, A., Seyfari, Y. and Lotfi, S. (2020). Android malware detection using feature selection with hybrid genetic algorithm and simulated annealing. In Proceedings of the 2020 IEEE 5th Conference on In Electrical and Computer Engineering (ETECH) Information and Communication Technology (ICT),1- 7.

Niyomubyeyi, O., Sicuaio, T. E., Díaz González, J. I., Pilesjö, P. and Mansourian, A. (2020). A comparative study of four metaheuristic algorithms, AMOSA, MOABC, MSPSO, and NSGA-II for evacuation planning. Algorithms, 13 (1), 16.

Senanayake, J., Kalutarage, H. and Al-Kadri, M. O. (2021). Android mobile malware detection using machine learning: A systematic review. Electronics, 10(13), 1606.

Statista Research Department (2023). Statista. Available from: https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/. (Accessed on 1 November 2022).

Storn, R. and Price, K. (1997). Differential Evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, (11), 341-359.

Şahin, B.C. (2022). Learning optimized patterns of software vulnerabilities with the clock-work memory mechanism. European Journal of Science and Technology, (41), 156-165.

Tahtacı, B. and Canbay, B. (2020). Android malware detection using machine learning. Innovations in Intelligent Systems and Applications Conference (ASYU), 1-6.

Ullah, A., Şahin, B.C., Dinler, Ö.B., Khan, M.H., and Aznaoui, H. (2021). Heart disease prediction using various machines learning approach. Journal of Cardiovaskular Disease Research, 3(12), 379-391.

Van Laarhoven, P.J. and Aarts, E.H. (1987). Simulated Annealing. Springer Netherlands, 7-15.

Waleed, A. (2019). Hybrid intelligent Android malware detection using evolving support vector machine based on genetic algorithm and particle swarm optimization. IJCSNS International Journal of

ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI:10.29132/ijpas.1382344

Computer Science and Network Security, 9 (19), 15-28.

Wang, L., Gao, Y., Gao, S. and Yong, X. (2021). A new feature selection method based on a self-variant genetic algorithm applied to android malware detection. Symmetry, 13 (7), 1290.

Yerima, S. Y. and Sezer, S. (2018). Droidfusion: A novel multilevel classifier fusion approach for android malware detection. IEEE transactions on cybernetics, 49 (2), 453-466.

Yildiz, O. and Doğru, I.A. (2019). Permission-based android malware detection system using feature selection with genetic algorithm. International Journal of Software Engineering and Knowledge Engineering, 29 (2), 245–262.

Zhou, Y. and Jiang, X. (2012). Dissecting android malware: characterization and evolution. In 2012 IEEE symposium on security and privacy, IEEE, 95-109. Available from: http://www.deeplearningbook.org (Accessed on 1 November 2022).