The multi-objective feature selection in Android malware detection system

Anahita Golrang[1], Sule Yildirim Yayilgan[2], and Ogerta Elezaj[2]

[1] Department of Computer Science, Norwegian University of Science and Technology, 2815 Gjøvik, Norway Anahitam@stud.ntnu.no
[2] Department of Information Security and Communication Technology, Norwegian University of Science and Technology, 2815 Gjøvik, Norway
sule.yildirim@ntnu.no ; ogerta.elezaj@ntnu.no

Abstract. The Android operating system boasts a global market share over the previous years, which has made it the most popular operating system in the world. Recently, Android has become the target of attacks by cybercriminals because of its open-source code and its progressive growth. Many machine learning techniques have been used to address this issue in the Android operating system. However, a limited range of feature selection methods has been used in these systems. This paper, therefore, aims to address and evaluate the impact of a multi-objective feature selection approach called NSGAII in Android malware detection systems. To improve the diversity of solutions offered by this method, we have modified the standard NSGAII approach. Experimental results show that the proposed method can lead to better malware classification.

Keywords: Feature selection · Android malware detection system· multi-objective optimization

1 Introduction

Android has dominated the mobile device industry. Just in 2019, the percentage of mobile devices using the Android operating system was around 76 % [1]. This operating system is supported by various types of applications in different markets which provides grateful functionality to its users. However, the total number of malware applications on the android market has boomed catastrophically, due to being an open-source software and also being one of the most used mobile operating systems. Moreover, many malicious applications are found on unofficial Android markets, where security issues receive less attention. Furthermore, as mentioned formerly, the market share for Android is considerably high compared to other operating systems. Therefore, Android is the hot target of attackers to gain more control of the system. The machine learning approaches have been proposed in recent years as one of the effective solutions in Android malware detection systems [2]. However, data mining researchers consider high-dimensional data analysis as a challenge. Feature selection known as the process of selecting a subset of features that contribute most to the classification process

in machine learning activities brings many advantages in the reduction of the data dimensionality and enhancement of classifier efficiency [3]. The effectiveness of these methods has been established in improving the learning performance as well as the high- dimensional data processing. The feature selection technique aims at reducing the irrelevant features, reducing the computation cost, increase prediction performance and gaining a better understanding or representation of the data [4]. Although feature selection methods have been used widely in other fields such as intrusion detection systems [5], the majority of papers in the area of Android malware detection tend to select the essential features by rationalizing which require a deep understanding of the nature of the features involved in Android applications. The research in [2] and [6] have provided review papers on various types of features which have been applied in Android malware detection systems. The results of their investigations demonstrate that despite the advantages of feature selection approaches they have been applied in around 8% to 13% of the reviewed papers. To the best of our knowledge, the majority of feature selection approaches currently used in Android malware detection systems encounter difficulties concerning the nature of the problem. The feature selection problem can be modelled as two conflicting objectives which are to minimize the number of features while offering a higher classification performance (minimize classification error). The single objective feature-selection approaches applied in Android malware detection systems are not able to confront both objectives simultaneously while the multi-objective techniques could be considered as a potential solution to the aforementioned issue [7]. Hence, we would attempt to assess the impact of multi-objective feature selection techniques compared to single-objective approaches in Android malware detection systems. To achieve this objective, the following research questions are defined :

1. Can multi-objective techniques perform better compared to single-objective feature selection approaches in Android malware detection systems?
2. Is there any special deficiency in current multi-objective approaches which could be improved?

Consequently, the gap mentioned is hoped to be filled by using multi-objective techniques as a feature selection method. To address the first question, the proposed framework applies the multi-objective genetic algorithm, namely NSGAII, for the feature selection purpose. The research done in [8] introduces the redundancy issue as one of the intrinsic issues in NSGAII algorithm. Hence, a modified version of the NSGAII method which removes the redundant solutions have been proposed in this paper which has been tested on two well-known datasets in the field of Android malware detection systems. The most significant contributions of this paper can be folded as follows:

1. To the best of our knowledge, there is no other research in the literature assessing the impact of the multi-objective feature selection approaches in malware detection in Android systems.
2. The multi-objective feature selection technique used in this research is a modified version of the standard NSGAII method, which improves the diversity of the solutions.

The rest of this paper is constructed as follows, wherein Section 2, we give a brief overview of related works. Section 3 describes the methodology used in the proposed Android malware detection system by defining the dataset in use, the modified feature selection, and the malware detection approach, respectively. The experimental results are discussed in Section 4, where the evaluation metrics are defined prior to the results achieved in the experiments. Finally, the conclusions of the research are outlined in Section 5.

2 Background

As this study is aimed at improving feature selection approaches used in Android malware detection systems, the literature is explored on this subject. Coronado-De-Alba et al. [9] examined the impact of two different feature selection methods called chi-square and relief approach. The malware samples in this dataset are derived from the Drebin dataset. However, the benign samples are gathered from Google Play, and third-party stores after evaluation by the Virustotal tool [10]. The features used in this experiment are divided into permissions, intents, hardware, and software categories. They have introduced Random Forest and random committee as the most efficient meta learner classifiers. In the next step, Random Forest with 200 random trees is selected while using the random committee as the base classifier, which showed the best outcomes on an unbalanced dataset and without feature selection. However, they report these feature selection methods with the ensemble of classifiers to show better efficiency compared to single classifiers. As a result, they suggest these methods in situations where velocity plays a vital role to cope with the dataset size. Zhao et al. [11] proposed a novel feature selection method called FrequenSel. In this research, more than 32000 features have been gathered before feature selection. These features are divided into APIs, permissions, actions, IP, and URLs. They have mentioned the distribution bias in favor of benign apps features and the long-tail effect as the main issues regarding traditional feature selection methods such as information-gain and chi-square. To solve this problem, the features selected by their algorithm should follow two conditions. First of all, its usage percentage should be higher in malware compared to benign samples. Next, a threshold has been introduced that ensures the proper occurrence of features in all malware samples. The proposed framework depicts better results compared to information gain and chi-square. The research done in [12] has proposed a novel architecture in Android malware detection systems. They have gathered permissions, and API function calls from the Android app samples. In the experiments, the benign app is gathered by Google Play, and the malware data are taken from the Malgenome project. The results of their experiments show that the API calls have a higher impact on the efficiency of the models compared to the permission features. As a result, they have selected a higher range of API calls after the feature selection process. In the feature selection step, ANOVA and SVM-RFE approaches have been applied to rank the features. Finally, they

have chosen the first 300 API calls features in the sorted list as well as the top 80 features in the permission set.

3 Methodology

In this section, the datasets used to evaluate the proposed method are described initially. Next, we have defined the feature selection approach and the two-step modifications applied to the feature selection technique, and we would explain how These modifications are done to improve the diversity of the solutions. Finally, the malware detection approach, which divides the samples into benign and malware categories, is discussed in detail.

3.1 Dataset

The experiments have been conducted on two datasets consists of real-world Android application samples, details of which are presented in Table 1. Each of this dataset contains 215 features. We have selected these datasets as they are widely used in the research community. During the dataset generation process, AXMLprinter2 is used to extract permissions and intents from the manifest file. Moreover, the extra features in API calls are derived from the .dex files using reverse engineering by Baksmali disassembler. Finally, the most important ad libraries introduced in [13] have been excluded to improve the quality of API call feature extraction.

Table 1. The sample distributions in Drebin [14], and Malgenome [15] datasets

Dataset	#Samples	#malware	#benign	#features
Malgenome-215	3799	1260	2539	215
Drebin-215	15036	5560	9476	215

3.2 Feature selection

In order to remove the irrelevant features, we have applied feature selection approaches which lead to the dimensionality reduction of the datasets and potential improvement of the classification performance. As mentioned previously, the majority of the projects in this field of study have chosen the list of features based on rationalizations which requires a deep understanding regarding the nature of features available in the datasets. To the best of our knowledge, the multi-objective approaches have not been used for feature selection purposes in Android malware detection systems. This provides a gap for further investigation. In our previous research [5], we proposed an intrusion detection framework based on a modified multi-objective feature selection approach called NSGAII.

Hence, it provides enough motivation to assess the effectiveness of this method for feature selection purpose in Android malware detection systems. The modification which was applied to the aforementioned research was improving the diversity of solutions by removing the identical redundant solutions. In this research, two levels of modification have been applied to the standard NSGAII technique. Algorithm 1 demonstrates the steps involved in the modification process where the duplicate solutions referring to exactly the same feature sets are removed in the first level. In the second step, the solutions referring to various feature sets are evaluated to ensure that they report distinct objective functions. In the following paragraphs, we would describe the modification process in detail.

Algorithm 1: MODIFIED NSGAII technique

```
Input: pop population
Output: Rank(S) for each solution S in pop
C = the array from CFS method
Step 1. Remove all duplicate solutions in the initial population
Step 2. for each solution i and j do
    if objective function (i) = objective function (j) then
        A= correlation (i, C) and B=correlation (j, C)
        if  if (A < B) then
            
        remove A
        else
            remove B
```

Modified NSGAII

Step 1: The presence of duplicate solutions in standard NSGAII techniques have been mentioned as a significant threat to diversity [16] and convergence speed [8] of this algorithm. In our previous research [5], we modified the NSGAII technique to remove the duplicate solutions in the merged population. Hence, we applied the same modification technique to remove the duplicate solutions and improve the diversity and convergence speed of the NSGAII algorithm in the first modification step in this research as well.

Step 2: Although in step 1, we have removed the identical solutions referring to the exactly the same solutions, there is still the possibility that the solutions with other degrees of similarity are still present in the offered solutions. In Step 2,

we would attempt to remove the solutions reporting exactly the same objective-functions since we would consider these solutions as redundant solutions that their presence is not adding any extra value to the quality of the proposed solution, and this fact provides the motivation to apply the modification in the second step. To ascertain which of the overlapping solutions may provide higher value, we have attempted to discover which one has the highest similarity with a single objective method available in the Weka environment, namely CFS. The reason for choosing a feature selection method is the fact that these solutions are referring to various feature sets proposed by the NSGAII technique. It should be mentioned that the CFS technique could be replaced by any other feature selection approach available in the state-of-the-art. To achieve the highest similarity degree, the linear association between each of the overlapping solutions (i and j) and the array achieved by CFS is calculated using the correlation-coefficient concept. The linear association between the two- vectors , $A_1 = (A_1, A_2, \ldots, A_n)$ and $B = (B_1, B_2, \ldots, B_n)$, is called the Pearson correlation, and can be calculated according to Equation 1.

$$correlation(A, B) = \frac{\sum_{i=1}^{n}(A_i - \overline{A})(B_i - \overline{B})}{\sqrt{\sum_{i=1}^{n}(A_i - \overline{A})^2}\sqrt{\sum_{i=1}^{n}(B_i - \overline{B})^2}} \tag{1}$$

Finally, the solution which reports higher degree of correlation using equation 1 is chosen and the other one is eliminated from the list of offered solutions.

3.3 Malware detection

In the malware detection step, we would like to divide the solutions into a binary classification of malware and benign categories. The feature selection technique defined in section 3.2 provides a list of solutions which provide various feature subsets. Among all of these solutions, we would select the solution with minimum f_2 value which is equivalent to the error rate reported for the corresponding feature set. Next, we would reduce the size of the dataset according to the proposed list of feature sets. Afterwards, the Random Forest classifier, as an ensemble method and peered review machine learning technique found in literature [9], would be applied on the datasets to evaluate the efficiency of the proposed solution and divide the samples into benign and malware categories.

4 Experimental Results

In this section, we would initially describe the evaluation metrics used to assess the effectiveness of the proposed feature sets. Afterwards, the results of two experiments on Drebin and Malgenome datasets are reported according to the aforementioned metrics.

4.1 Evaluation metrics

Weighted F Measure is applied in the research conducted in [17] to evaluate the efficiency of the method. The dataset used in this research is extracted from this research. Hence, we apply the same metric as described in equation 2 by WFM.

$$WFM = \frac{(F_m * N_m) + (F_b * N_b)}{(N_m + N_b)} \tag{2}$$

Where the F_m, F_b, N_m, and N_b used in the evaluation process refer to the F Measure and number of instances in malware and benign instances, respectively. The F value in this equation correlates with the F Measure value where it can be calculated according to the equation 3.

$$F\ Measure = 2 * (\frac{Precision * Recall}{Precision + Recall}) \tag{3}$$

While the Precision and Recall are formulated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

Where TP, TN, FP, and FN are equal to the following values:

- **True Positive (TP):** the number of correctly classified malware files.
- **True Negative (TN):** the number of correct classification of benign samples as benign.
- **False Negative (FN):** Malicious applications wrongly classified as benign .
- **False Positive (FP):** the number of misclassified legitimate applications .

4.2 Results

This section describes the experimental results obtained with the improved NASGII over the datasets selected for these experiments. The proposed feature selection solution was implemented using Matlab R2019a. Next, the Random Forest algorithm is applied to the reduced dataset in the Waikato Environment for Knowledge Analysis (Weka 3.8). The data analyses are performed using a PC with Intel Core i7 processor, 2.1 GHz speed and 8 GB RAM. To evaluate the performance of each model, a 10-fold cross-validation resampling method was applied. This method is selected for the advantages it has in guaranteeing the independence of the validation set from the training one and the validity of the trained classifier against any unseen sample.

Experiment one on Drebin dataset: Figure 1 demonstrates the feature subset solutions proposed by the modified multi-objective NSGAII. Each $*$ symbol in this fig is considered as the demonstration of an individual solution. Further detail about these solutions can be found in Table 2 where the Num, NF, MSE, and ratio refer to the order of solution on the Pareto front, the number of selected features, Mean-square-error (MSE) of the solution achieved by the artificial neural network (ANN), and the percentage of the selected features in each solution compared to the full features available in the original datasets. We have chosen the last solution with the minimum MSE value, which corresponds to 186 features of total 215 features available in Drebin dataset.

Table 2. Pareto Front in Drebin

Num	NF	MSE	Ratio
1	81	0.023	37%
2	84	0.023	39%
3	87	0.021	40.4%
4	88	0.019	40.9%
5	89	0.018	41%
6	93	0.018	43.2%
7	94	0.018	43.7%
8	95	0.017	44%
9	97	0.016	45%
10	100	0.016	46.5%
11	101	0.013	46.9%
12	112	0.013	52%
13	117	0.013	54%
14	122	0.011	56%
15	121	0.012	56.2%
16	168	0.011	78%
17	186	0.0094	86%

Table 3. Pareto Front in Malgenome

Num	NF	MSE	Ratio
1	78	0.019	36%
2	80	0.018	37.2%
3	81	0.010	37.6%
4	83	0.005	38%
5	88	0.005	40%
6	90	0.004	41%
7	91	0.004	42%
8	94	0.0042	43%
9	105	0.0040	48%
10	106	0.003	49%
11	113	0.003	52%
12	115	0.002	53%

Next, the features unavailable on the feature subset solution are eliminated from the dataset, and the dataset is fed into the malware detection step where the Random Forest technique available in Weka environment would classify the samples into malware and benign categories. Table 4 demonstrates the confusion matrix derived from the classification approach on Drebin dataset. We have compared the proposed method with two single objective feature- selection techniques called CFS and FilterSubsetEval. To achieve a uniform structure for comparison, we have initially applied each of these techniques on Drebin dataset. Afterwards, the malware detection approach is applied to the reduced size datasets using Random Forest. Moreover, the proposed method results are compared to another research called Droid-Fusion [17], in which they have published the final version of Drebin and Malgenome datasets used in our research. Table 5 reports the results of this comparison, where the proposed method shows

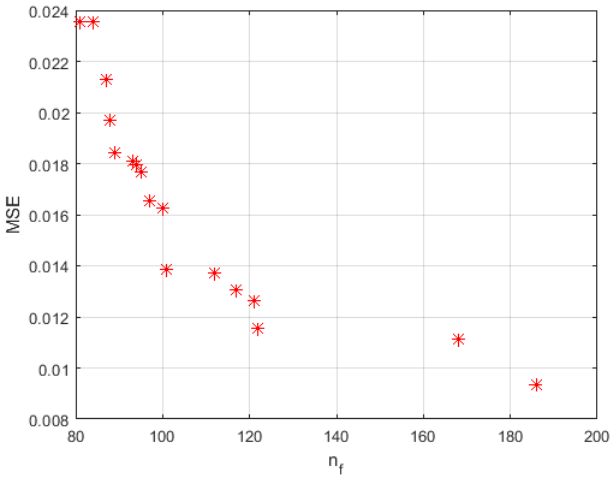

Fig. 1. Pareto front derived from Drebin

higher efficiency in all performance metrics compared to both single objective feature selection approaches. However, it demonstrates better results in three out of five metrics in comparison with Drebin. To evaluate the efficiency of the method in both benign and Malware samples, the precision and recall factors have been reported in malware and benign classes separately instead of a single average factor in all samples.

Table 4. Drebin confusion Matrix

	Malware	Benign	Total
Malware	5429	131	5560
Benign	39	9437	9476
Total	5468	9568	15036

Experiment two on Malgenome dataset: The proposed feature selection has been applied on Malgenome dataset as well. Figure 2 demonstrates the feature subset solutions offered by the proposed multi-objective approach in this research. Twelve distinct solutions have been offered by this technique where the solution with the least MSE rate has been chosen to be used to reduce the feature subset size of the Malgenome dataset. Table 3 demonstrates the details of the offered solutions by the multi-objective technique. We have chosen the last solution in this table where 115 features out of 215 features available on Malgenome dataset have been chosen which is equivalent to 53% of the total fea-

Table 5. Evaluation measures for Drebin dataset

	Pre-M	Rec-M	Prec-B	Rec-B	W-FM
Droid-Fusion [17]	0.981	0.984	0.991	0.989	0.9872
CFS-Random Forest	0.931	0.927	0.957	0.959	0.949
FilterSubsetEval-Random Forest	0.931	0.927	0.957	0.959	0.949
Proposed-Method	**0.993**	**0.976**	**0.986**	**0.996**	**0.989**

tures. Next, the reduced size dataset is classified using Random Forest classifier.

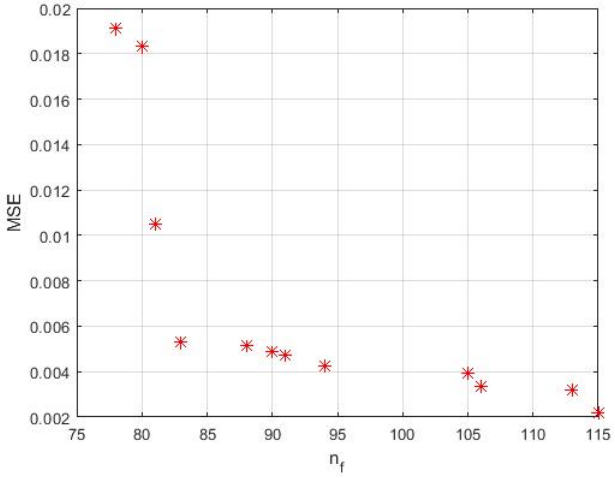

Fig. 2. Pareto front derived from Malgenome

The confusion matrix derived from this process can be found in table 6. We have compared the proposed method with the same state-of-the-art mentioned in the Drebin experiment. The evaluation results of our experiment demonstrate the superiority of the proposed multi-objective feature selection technique compared to two single objective feature selection methods, namely CFS and FilterSubsetEval in all factors and its improvement in three out of five metrics compared to Droid-Fusion [17].

Table 6. Malgenome confusion Matrix

	Malware	Benign	Total
Malware	1239	21	1260
Benign	10	2529	2539
Total	1249	2550	3799

Table 7. Evaluation measures for Malgenome dataset

	Pre-M	Rec-M	Prec-B	Rec-B	W-FM
Droid-Fusion [17]	0.984	0.968	0.984	0.992	0.984
CFS-Random Forest	0.968	0.948	0.974	0.985	0.972
FilterSubsetEval-Random Forest	0.975	0.980	0.960	0.988	0.978
Proposed-Method	**0.992**	**0.983**	**0.992**	**0.996**	**0.992**

5 Conclusion

By selecting the relevant features and by applying different machine learning algorithms, it is possible to effectively detect Android malware. The aim of this study was to explore the use of a multi-objective function for feature-selection purpose on Android malware detection systems in order to create the optimal feature subsets. Hence, a NSGAII-ANN method is applied to construct the basis for the feature-selection stage. To improve the proposed framework, we have modified the traditional NSGAII method to solve one of the most significant deficiencies in NSGAII algorithm by a redundant solution removal approach in two stages. Moreover, the Random Forest method is applied in order to evaluate the selected subsets. Experimental results show that the proposed method is superior to other methods found in literature in terms of improving the classification metrics for both datasets. In the future, we would like to test our proposed solution on other real datasets covering a broader range of malware, and we would like to apply and compare different machine learning algorithms. As Android malware detection techniques are mostly applied on imbalanced network traffic datasets, different experiments should be performed applying oversampling and downsampling techniques, expecting to achieve higher accuracy on both the minority class and the entire datasets.

References

1. Mobile Operating System Market Share Worldwide. https://gs.statcounter.com/os-market-share/mobile/worldwide. Accessed on 2020-04-13.
2. Feizollah, A., Anuar, N.B., Salleh, R. and Wahab, A.W.A., 2015. A review of feature selection in mobile malware detection. Digital investigation, 13, pp.22-37.
3. Cai, J., Luo, J., Wang, S. and Yang, S., 2018. Feature selection in machine learning: A new perspective. Neurocomputing, 300, pp.70-79.

4. Chandrashekar, G. and Sahin, F., 2014. A survey on feature selection methods. Computers & Electrical Engineering, 40(1), pp.16-28.
5. Golrang, A., Golrang, A.M., Yayilgan, S.Y. and Elezaj, O., 2020. A Novel Hybrid IDS Based on Modified NSGAII-ANN and Random Forest. Electronics, 9(4), p.577.
6. Wang, W., Zhao, M., Gao, Z., Xu, G., Xian, H., Li, Y. and Zhang, X., 2019. Constructing features for detecting android malicious applications: issues, taxonomy and directions. IEEE Access, 7, pp.67602-67631.
7. Xue, B., Zhang, M., Browne, W.N. and Yao, X., 2015. A survey on evolutionary computation approaches to feature selection. IEEE Transactions on Evolutionary Computation, 20(4), pp.606-626.
8. Huang, B., Buckley, B. and Kechadi, T.M., 2010. Multi-objective feature selection by using NSGA-II for customer churn prediction in telecommunications. Expert Systems with Applications, 37(5), pp.3638-3646.
9. Coronado-De-Alba, L.D., Rodr´ıguez-Mota, A. and Escamilla-Ambrosio, P.J., 2016, November. Feature selection and ensemble of classifiers for Android malware detection. In 2016 8th IEEE Latin-American Conference on Communications (LATINCOM) (pp. 1-6). IEEE.
10. VIRUSTOTAL. https://www.virustotal.com/gui/home/upload. Accessed on 2020-04-13
11. Zhao, K.; Zhang, D.; Su, X.; Li, W. Fest: A feature extraction and selection tool for Android malware 373 detection. 2015 IEEE symposium on computers and communication (ISCC). IEEE, 2015, pp. 714–720
12. Qiao, M., Sung, A.H. and Liu, Q., 2016, July. Merging permission and API features for Android malware detection. In 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI) (pp. 566-571). IEEE.
13. Cen, L., Gates, C.S., Si, L. and Li, N., 2014. A probabilistic discriminative model for android malware detection with decompiled source code. IEEE Transactions on Dependable and Secure Computing, 12(4), pp.400-412.
14. Arp, D., Spreitzenbarth, M., Hubner, M., Gascon, H., Rieck, K. and Siemens, C.E.R.T., 2014, February. Drebin: Effective and explainable detection of android malware in your pocket. In Ndss (Vol. 14, pp. 23-26).
15. Zhou, Y. and Jiang, X., 2012, May. Dissecting android malware: Characterization and evolution. In 2012 IEEE symposium on security and privacy (pp. 95-109). IEEE.
16. Chaiyaratana, N., Piroonratana, T. and Sangkawelert, N., 2007. Effects of diversity control in single-objective and multi-objective genetic algorithms. Journal of Heuristics, 13(1), pp.1-34.
17. Yerima, S.Y.; Sezer, S. DroidFusion: A Novel Multilevel Classifier Fusion Approach for Android Malware 384 Detection. IEEE Transactions on Cybernetics 2018.