Ahmad FIRDAUS, Nor Badrul ANUAR, Ahmad KARIM, Mohd Faizal Ab RAZAK, 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. https://doi.org/10.1631/FITEE.1601491

Discovering optimal features using static analysis and a genetic search based method for Android malware detection

Key words: Genetic algorithm; Static analysis; Android; Malware; Machine learning

Main corresponding author: Ahmad FIRDAUS
E-mail: ahmadfirdaus@um.edu.my

ORCID: https://orcid.org/0000-0002-7116-2643

Motivations

1. Mobile device manufacturers are rapidly producing miscellaneous Android versions worldwide.
2. Simultaneously, cyber criminals are executing malicious actions, such as tracking user activities, stealing personal data, and committing bank fraud. These criminals gain numerous benefits as too many people use Android for their daily routines, including important communications.
3. Detecting unknown malware specifically on Android with minimal number of features is always a challenging task.
4. The advantages of static analysis that covers overall code, low resource consumption, and rapid processing.

Main ideas

1. We investigated and identified 106 features in four categories (i.e. permission, code-based, directory path, and system command) and their existence in Drebin malware dataset.

2. To the best of our knowledge, this is the first time that genetic search (GS) has been used in Android static analysis studies for selecting features. This method was applied to select the best among the 106 features for detecting malware using machine learning.

3. To compare both regularly and infrequently used machine learning classifiers. Moreover, we selected 5 types of classifiers to compare the results for different categories, namely, Naïve Bayes (NB), functional trees (FT), J48, random forest (RF), and multilayer perceptron (MLP).

Methods

1. GS selected the best features in permission, directory path, and code-based categories which comprises of 106 features.

2. This study used an empirical evaluation involving five machine learning classifiers: Naïve Bayes (NB), multilayer perceptron (MLP), functional trees (FT), J48, and random forest (RF), to measure the effectiveness of the features selected by GS.

3. The GS is able to optimize the features from 106 to only 6 and achieve outstanding results.

Major results

1. In cross validation, FT marked good detection among other machine learning classifiers with different categories.

Table 15 Classifier results in cross validation

Classifier	Category	Accuracy	Number of instances		FPR	TPR	ROC	Precision	Recall	*F*-measure
			Correctly classified	Incorrectly classified						
Bayes	NB	94.17%	5749	356	24.50%	**96%**	0.936	97.50%	**96%**	**96.80%**
Function	MLP	94.19%	5750	355	24.20%	**96%**	**0.950**	**97.60%**	**96%**	**96.80%**
Trees	FT	**94.22%**	**5752**	**353**	**23.80%**	**96%**	0.947	**97.60%**	**96%**	**96.80%**
	RF	94.20%	5751	354	24%	**96%**	0.946	**97.60%**	**96%**	**96.80%**
	J48	93.89%	5732	373	25.10%	95.80%	0.905	97.50%	95.80%	96.60%

Major results

2. In training and testing benchmark, FT is the outstanding classifier, achieving the highest values in all categories except TPR and recalling in detecting unknown malware.

Table 16 Classifier results in cross validation

Classifier	Category	Accuracy	Number of instances		FPR	TPR	ROC	Precision	Recall	*F*-measure
			Correctly classified	Incorrectly classified						
Bayes	NB	**95%**	**1160**	**61**	22.10%	**96.80%**	0.940	97.70%	**96.80%**	**97.20%**
Function	MLP	94.90%	1159	62	22.10%	96.70%	0.953	97.70%	96.70%	**97.20%**
Trees	FT	**95%**	**1160**	**61**	**21.20%**	96.70%	**0.956**	**97.80%**	96.70%	**97.20%**
	RF	94.90%	1159	62	22.10%	96.70%	0.954	97.70%	96.70%	**97.20%**
	J48	94.70%	1157	64	23.90%	96.70%	0.916	97.50%	96.70%	97.10%

Major results

3. The proposed GS able to detect unknown malware in short amount of time.

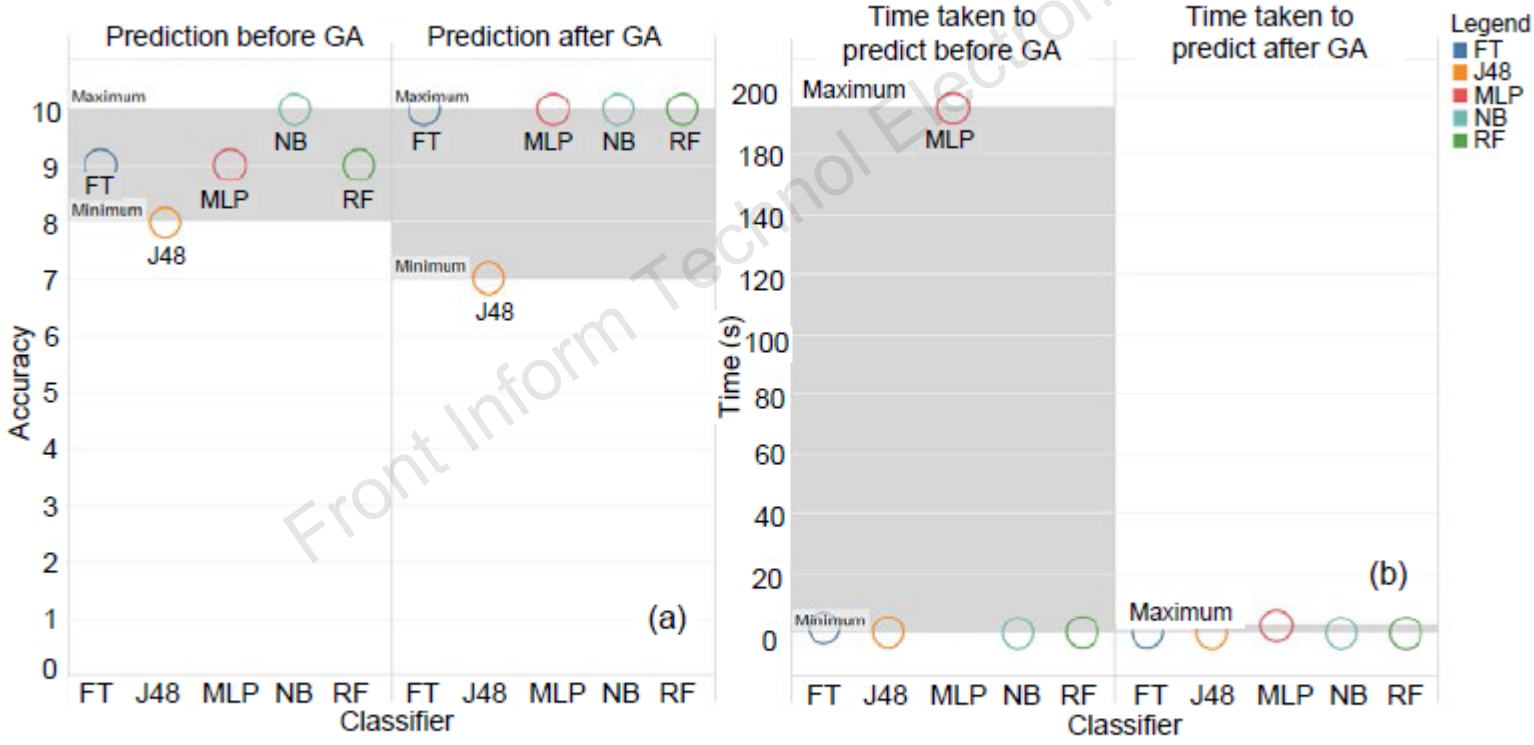

Fig. 13 A comparison of accuracy (a) and time consumption (b) in machine learning prediction

Conclusions

1. As features are crucial in static analysis, it is necessary to select the smallest number of the best features to enhance accuracy (i.e., to produce an accurate predictive model) with fewer data and to reduce model complexity.

2. GS is able to search the best among those 106 features (permission, code-based, directory path, and system command categories) and genetically selects six exclusive features.

Conclusions

3. Thereafter, we evaluate the features in five machine learning classifiers (i.e., NB, FT, J48, RF, and MLP). Among them, FT recorded the highest accuracy (94.22%) and TPR (96%), and the lowest FPR (23.8%) in 10-fold cross validation. FT achieved the highest scores in training and testing part, with 95% accuracy, 1160 correctly and 61 incorrectly classified instances, 21.2% FPR, 0.956 ROC, 97.8% precision, and 97.2% *F*-measure. The experiment showed that FT recorded magnificent scores.