Key features for the characterization of Android malware families

JAVIER SEDANO* and SILVIA GONZÁLEZ*, *Instituto Tecnológico de Castilla y León, C/ López Bravo 70, Pol. Ind. Villalonquejar, 09001, Burgos, Spain.*

CAMELIA CHIRA**, *Department of Computer Science, University of Cluj-Napoca, Baritiu 26-28, Cluj-Napoca 400027, Romania.*

ÁLVARO HERRERO†, *Department of Civil Engineering, University of Burgos, Avenida de Cantabria s/n, 09006, Burgos, Spain.*

EMILIO CORCHADO‡, *Department of Computer Science and Automation, University of Salamanca, Plaza de la Merced, s/n, 37008, Salamanca, Spain.*

JOSÉ RAMÓN VILLAR§, *Computer Science Department, University of Oviedo, ETSIMO 33005, Oviedo, Spain.*

Abstract

In recent years, mobile devices such as smartphones, tablets and wearables have become the new paradigm of user–computer interaction. The increasing use and adoption of such devices is also leading to an increased number of potential security risks. The spread of mobile malware, particularly on popular and open platforms such as Android, has become a major concern. This paper focuses on the bad-intentioned Android apps by addressing the problem of selecting the key features of such software that support the characterization of such malware. The accurate detection and characterization of this software is still an open challenge, mainly due to its ever-changing nature and the open distribution channels of Android apps. Maximum relevance minimum redundancy and evolutionary algorithms guided by information correlation measures have been applied for feature selection on the well-known Android Malware Genome (Malgenome) dataset, attaining interesting results on the most informative features for the characterization of representative families of existing Android malware.

Keywords: Feature selection, evolutionary computation, max-relevance min-redundancy criteria, information correlation coefficient, Android, malware.

1 Introduction

Since the first smartphones came onto the market in the late 90s, sales on that sector have increased constantly until present days. Among all the available operating systems, Google's Android is the most popular mobile platform, according to Statista [1]. The number of Android-run units sold in Q4

*E-mail: javier.sedano@itcl.es; silvia.gonzalez@itcl.es
**E-mail: camelia.chira@cs.utcluj.ro
†E-mail: ahcosio@ubu.es
‡E-mail: escorchado@usal.es
§E-mail: villarjose@uniovi.es

Vol. 25 No. 1, © The Author 2016. Published by Oxford University Press. All rights reserved.
For Permissions, please email: journals.permissions@oup.com
doi:10.1093/jigpal/jzw046 Advance Access published 5 August 2016

2015 (latest available data) worldwide raised to 325.39 million out of 403.12 million units, that is a share of 80.71%. Similarly, the number of apps available at Android's official store has increased constantly from the very beginning, up to more than 2 million [2] that are available nowadays. Moreover, Android became the top mobile malware platform as well. Some other statistics [3] confirm this trend as 99% of the new threats that emerged in Q1 2014 were run on Android. This operating system is an appealing target for bad-intentioned apps, reaching unexpected heights, as there are cases where PC malware is now being transfigured as Android malware [3] and the volume of Android malware spiked to 7.10 million in first half of 2015 [4] when it was 4.26 million at the end of 2014.

Smartphone security and privacy are nowadays major concerns. In order to address these issues, it is required to understand the malware and its nature. Otherwise, it will not be possible to practically develop an effective solution [5]. Thus, present study is not focused on the detection of Android malware but on the characterization of Android malware families instead. The proposed solution is related to the idea of reducing the amount of app features needed to distinguish among malware families. To do so, Malgenome (a real-life publicly-available) dataset [6] has been analyzed by means of several feature selection strategies. From the samples contained in such dataset, several alarming statistics were found [5], that motivate further research on Android malware:

- Around one-third (36.7%) of the collected samples leverage root-level exploits to fully compromise the security of the whole system.
- More than 90% turn the compromised phones into a botnet controlled through network or short messages.
- 45.3% of the samples have the built-in support of sending out background short messages (to premium-rate numbers) or making phone calls without user awareness.
- 51.1% of the samples harvested user's information, including user accounts and short messages stored on the phones.

To improve the characterization of the addressed malware, this study proposes the use of feature selection. In order to easily identify the malware family an app belongs to, the feature selection task is addressed using an evolutionary algorithm guided by information theory measures. Each individual encodes the subset of selected features using a binary representation. The evolutionary search process is guided by the crossover and mutation operators applied to the binary encoding, and a fitness function that evaluates the quality of the encoded feature subset. The initial results of the proposed approach were presented in Sedano *et al*. [7], where the fitness function used was either mutual information or the information correlation coefficient (ICC). This paper extends this previous work by further investigating the subset of features selected by a novel weighted fitness function that allows some control over the number of selected features. The idea is to obtain a relevance of the features assessed by the information measures in a weighted balance with the size of the feature subset, as very few features were selected by the different algorithms applied in the seminal work.

In present study, the selected features are also ranked using the Minimum-Redundancy Maximum-Relevance (MRMR) measure [8, 9]. The MRMR feature selection framework is a well-known filter method based on both a maximal relevance criteria and minimum redundancy criteria. Besides a good relevance, MRMR requires selected features to further be maximally dissimilar to each other. The MRMR method is used in present study to compare or confirm the subsets of selected features related to Android malware, as determined by the evolutionary approaches based on the considered fitness functions. The results obtained are extensively analysed, describing their relevance that probes the positive aspects of gaining deep knowledge of malware nature.

Up to now, a growing effort has been devoted to Android malware [10]. From an intrusion detection perspective, many machine learning algorithms have been applied to differentiate between legitimate and malicious Android apps, such as classifiers [11]–[14] and clustering [15]. Under a similar perspective, some other approaches based on knowledge discovery [16], visual inspection [17] and weighted similarity matching of logs [18] have been also proposed. Feature selection [13, 19] in general terms, and MRMR [20] more precisely, have been previously applied to analyse Android malware. Present study differentiates from previous work as feature selection based on MRMR is now applied from a new perspective, trying to ease the characterization of different Android malware families rather than distinguishing between legitimate and malicious apps.

The structure of the paper is as follows: the information theory measures and the MRMR feature selection method are described in Section 2, the proposed evolutionary approach to feature selection is presented in Section 3, the setup of computational experiments for the Android Malware Genome dataset is described in Section 4, the results obtained are discussed in Section 5 and the conclusions of the study are drawn in Section 6.

2 Feature selection methods

Feature selection methods are generally used to improve the performance of algorithms. Such methods can reduce the number of features considered in a classification task by removing irrelevant or noisy features [21, 22].

Filter methods perform feature selection independently from the learning algorithm while wrapper models embed classifiers in the search model [9, 23]. Filter methods select features based on some measures that determine their relevance to the target class without any correlation to a learning method.

On the other hand, wrapper models integrate learning algorithms in the selection process and determine the relevance of a feature based on the learning accuracy [24]. Population-based randomized heuristics are normally used to guide the search towards the optimal feature subset. Wrapper methods require a high computational time and present a high risk of overfitting [24] but they are able to model feature dependencies and the interaction of the search model with the classifier [25].

2.1 Information theory measures

The relevance of features can be characterized in terms of mutual information or correlation. Mutual information is widely used to define dependency of variables. This subsection presents these two relevant measures from information theory [26]: mutual information (I) and ICC.

Mutual information is a measure of statistical independence, and is defined as follows. Let X be a random variable and $p(x)$ the probability distribution of X. Defined by means of their probability distribution, the mutual information between two variables has a higher value for higher degrees of relevance between the two features.

The mutual information between two features is denoted by $I(X, Y)$ and is given by:

$$I(X,Y) = \iint p(x,y) * \log\left(\frac{p(x,y)}{p(x) * p(y)}\right) \mathrm{d}x\mathrm{d}y. \tag{1}$$

A high value of the mutual information between a feature and a class means that the feature contains considerably information about the class. In feature selection problems, mutual information can be

used to determine the optimal feature subset by selecting those features with higher values of this measure.

The entropy $\mathrm{H(X)} = -\int p(x)\cdot\log(p(x))\mathrm{d}x$ is a measure of the information the feature supports. Similarly, $H(Y|X)$ denotes the entropy of a feature y provided the feature X. The ICC—$\mathrm{ICC}(X,Y)$—is calculated based on Equation (2) for a feature X and the output Y. ICC measures how independent two features are from each other (the higher the ICC value the more relevant the relationship is). The ICC measure is reflexive, symmetric and monotonic.

$$\mathrm{ICC}(X,Y) = \frac{I(X,Y)}{H(Y|X)} \tag{2}$$

If $\mathrm{ICC}(X,Y) = 1$ then the two variables X and Y are strictly dependent whereas a value of 0 indicates that they are completely irrelevant to each other. A correlation degree can be expressed by stating that X is relevant to Y with a degree coefficient of $\mathrm{ICC}(X,Y)$.

2.2 The MRMR method

The MRMR [9] feature selection method is a well-known filter method that obtains the maximum relevance to output and, at the same time, the minimum redundancy between the selected features.

In the first step, the MRMR approach selects one feature out of the N input features in the set X which has the maximum value of $I(X,Y)$. Let this feature be x_k. Next, one of the features in $X - x_k$ is chosen according to the MRMR criteria.

Let us suppose that we have $m-1$ features selected already in the subset S_{m-1} and the task is to select the mth feature from $X - S_{m-1}$. This will be the feature that maximizes the following formula:

$$max_{x_j \in X - \mathrm{S_{m-1}}} \left[\mathrm{I}\left(\mathrm{x_j}, \mathrm{y}\right) - \frac{1}{\mathrm{m}-1} \sum_{\mathrm{x_i} \in \mathrm{S_{m-1}}} \mathrm{I}\left(\mathrm{x_j}, \mathrm{x_i}\right) \right]. \tag{3}$$

This MRMR scheme can be run for $m = 1, 2, 3 \ldots$ resulting in different feature subsets.

3 Evolutionary approaches to feature selection

Since the number of features to be analysed in present study is small (26), the various feature subsets can be extensively evaluated using different methods. The results of these methods can then be aggregated in a ranking scheme. It is proposed to determine an ordered list of selected features using (i) a genetic algorithm (GA) based on information theory measures as fitness function and (ii) MRMR criteria [9].

Evolutionary algorithms are population-based search methods based on a certain individual representation, variation and selection operators as well as a fitness function that should be able to efficiently guide the search process. For the considered problem, a GA is developed based on the binary representation of selected features. The size of each individual equals the number of features (denoted by N) and the value of each position can be either 0 or 1, where 1 means that the corresponding feature is selected. The variation and selection mechanisms are standard.

It is proposed to evaluate the feature subset resulting from each individual by using I and ICC. Two GA variants are therefore developed, one corresponding to each one of the measures to calculate

the fitness of an individual. We denote this model by GA-INFO, where INFO can be either I or ICC (as described in [7]).

The GA-INFO algorithm is outlined below [7]. The population size is denoted by N, the maximum number of generations is denoted by G and t represents the current generation.

Algorithm: GA-INFO Feature Selection
Require: X the input variables data set
Require: Y the output vector
P ← a vector of N Individual objects
t ← 0
Generate the initial population P(t): randomly initialize the value of each individual
while t <G do
    Evaluate each individual IND in P(t): calculate I(IND, Y) or ICC(IND, Y) value
    P(t +1) ← roulette wheel selection from P(t)
    for all individuals IND in P(t + 1) **do**
        Select mate J from P(t + 1)
        K ←two-point crossover (IND, J)
        if fitness(K) >fitness(IND) **then**
            IND ← K
        end if
        L ← mutation(IND)
        if fitness(L) >fitness(IND) **then**
            IND ← L
        end if
    end for
    t ← t+1
end while
Return Best Individual in P(t)

The GA follows a standard scheme in which roulette wheel selection, two-point crossover and swap mutation are used to guide the search. Each individual is evaluated based on the correlation between the current subset of selected features and the output. This correlation is given by either I or ICC used to evaluate the fitness. Therefore, depending on the fitness function used, two GA-INFO variants are defined: GA−I is the GA using mutual information as fitness, while GA-ICC denotes the GA based on ICC fitness function.

Furthermore, a second variant of each GA-INFO (called GA-INFO-W, where W stands for *weighted*) is proposed in order to control the number of features selected in an individual. The fitness function of GA-INFO-W is based on a weighted scheme between the information theory measure and the number of selected features.

Let $k(x)$ be the size of the feature subset encoded in an individual x and w a real parameter between 0 and 1, denoting the weight of each fitness component. The weighted fitness function for an individual x is depicted in Equation (4).

$$f(x) = w \cdot \mathrm{INFO}(x) + (1 - w) \cdot 1/k(x). \tag{4}$$

The maximization of f would also lead to a minimum number of possible selected features in the individual. It should be noted that the features would only be selected as long as a high value of INFO(x) (either I or ICC) still emerges in the current individual. This balance is ensured by the value of the weight parameter w. A value of 0.5 would translate to an equal importance between the

information theory measure and the number of selected features. A higher value of$w$ can be used to give a relative higher importance to the information theory measure value compared with the size of the feature subset.

Similarly to the two GA-INFO variants (GA-I and GA-ICC), the GA-INFO-W model results in two variants called GA-I-W and GA-ICC-W.

4 Experiment setup

The experiments are based on the analysis of the Malgenome dataset [5], coming from the Android Malware Genome Project [6]. This is the first large collection of Android malware (1,260 samples) that was split in different malware families (49 in total). It covers the majority of existing Android malware, collected since their debut in August 2010.

Data related to many different apps were accumulated over more than 1 year from a variety of Android markets, and not only Google Play. Additionally, malware apps were thoroughly characterized based on their detailed behaviour breakdown, including the installation, activation and payloads.

Collected malware was split in 49 families, that were obtained by 'carefully examining the related security announcements, threat reports, and blog contents from existing mobile antivirus companies and active researchers as exhaustively as possible and diligently requesting malware samples from them or actively crawling from existing official and alternative Android Markets' [5]. The defined families were: ADRD, AnserverBot, Asroot, BaseBridge, BeanBot, BgServ, CoinPirate, Crusewin, DogWars, DroidCoupon, DroidDeluxe, DroidDream, DroidDreamLight, DroidKungFu1, DroidKungFu2, DroidKungFu3, DroidKungFu4, DroidKungFuSapp, DoidKungFuUpdate, Endofday, FakeNetflix, FakePlayer, GamblerSMS, Geinimi, GGTracker, GingerMaster, GoldDream, Gone60, GPSSMSSpy, HippoSMS, Jifake, jSMSHider, Kmin, Lovetrap, NickyBot, Nickyspy, Pjapps, Plankton, RogueLemon, RogueSPPush, SMSReplicator, SndApps, Spitmo, TapSnake, Walkinwat, YZHC, zHash, Zitmo and Zsone. Samples of 14 of the malware families were obtained from the official Android market, while samples of 44 of the families came from unofficial markets.

Information on those malware families is considered in present study. Thus, the analysed dataset consists of 49 samples (one for each family) and each sample has 26 different features. The features are divided into six categories; installation (repackaging, update, drive-by download, standalone), activation (BOOT, SMS, NET, CALL, USB, PKG, BATT, SYS, MAIN), privilege escalation (exploid, RATC/zimperlich, ginger break, asroot, encrypted), remote control (NET, SMS), financial charges (phone call, SMS, block SMS) and personal information stealing (SMS, phone number, user account). The values of those features are 0 (that feature is not present in that family) and 1 (the feature is present).

5 Computational results

The mutual information and information correlation between each feature and output is depicted in Figure 1. The features are given on the horizontal axis while the vertical axis gives the values of ICC and I, respectively, for each feature.

The MRMR method and the proposed GA-INFO and GA-INFO-W algorithms were used for the selection of the best features to characterize the considered Android malware families. The GA parameter setting used is the following: population size is 100, the number of generations is 100 and the number of runs for the algorithm is 50. The GA-INFO-W algorithms consider a value of 0.5 for

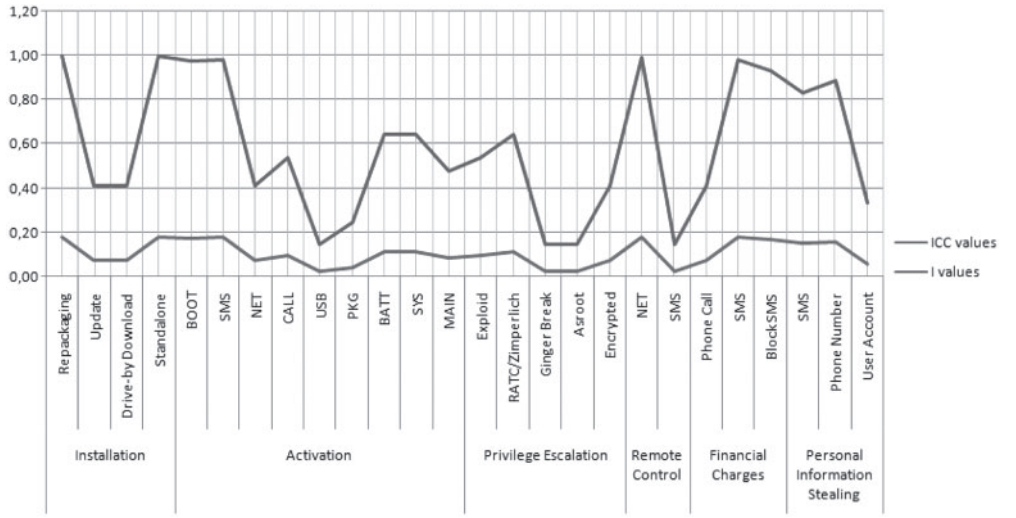

FIG. 1. ICC and I rates for each feature of the Malgenome dataset.

TABLE 1. Selected features by MRMR for different values of m

Feature	$m=2$	$m=3$	$m=4$	$m=5$
Installation—Repackaging	√	√	√	√
Activation—SMS	√	√	√	√
Activation—BOOT		√	√	√
Remote control—NET			√	√
Financial charges—SMS				√

the weight parameter, allowing a balanced search for relevant and in the same time small subset of features.

The MRMR method was used to select the m best features for $m=2/3/4/5$ features. Table 1 depicts the subset of features selected in each case. According to MRMR results, the most relevant features are Installation—Repackaging and Activation—SMS. When m was increased, the other features included in the relevant subset were Activation—BOOT, Remote Control—NET and Financial Charges—SMS.

GA-ICC and GA-I select two different individuals ([Installation—Repackaging, Installation—Standalone, Remote Control—NET, Activation—SMS] and [Installation—Repackaging, Installation—Standalone, Remote Control—NET, Financial Charges—SMS]), respectively, with the same fitness value for ICC and MI (0.18 and 0.99, respectively).

Figures 2 and 3 depict the evolution of GA-I and GA-ICC rates for of the 50 algorithm runs considered. Each line depicts one run and the best ICC from the population in each generation is depicted.

It should be noted that both GA-I and GA-ICC were able to reach the optimum values in the population very early in the search process—around generation 11. Each line represents a run of the algorithm and some lines overlap in some executions that were similar (see Figures 2 and 3). This

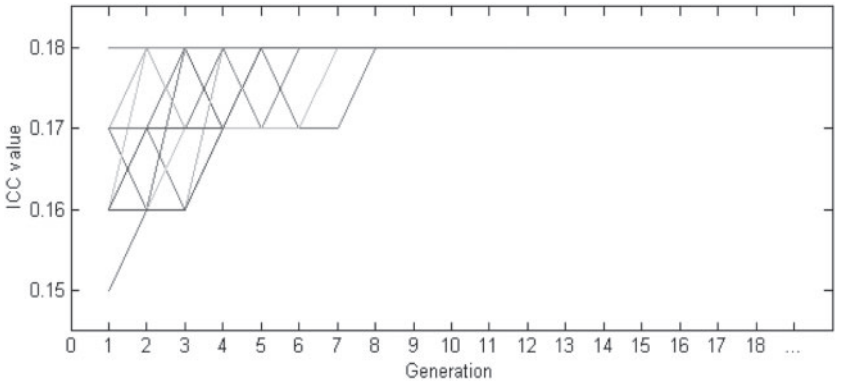

FIG. 2. Fitness ICC values in each generation of the 50 algorithm runs.

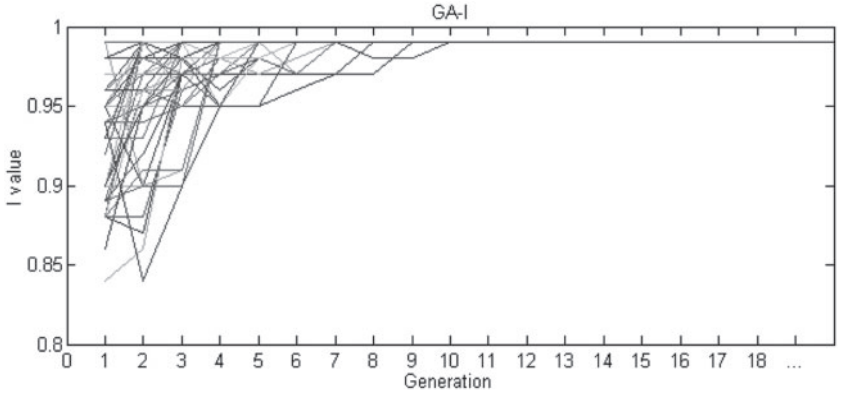

FIG. 3. Fitness I values in each generation of the 50 algorithm runs.

is due to the relatively small number of features that had to be considered in the search, leading to an individual size easy to handle and quickly explore many feature subsets.

The GA-I-W and GA-ICC-W models were applied for a weight value of 0.5 in order to allow balance between the information theory measure and the size of the feature subset. The evolution of fitness over the generations in each of the 50 runs is depicted in Figure 4 for GA-I-W and in Figure 5 for GA-ICC-W. The methods select only one feature in most individuals by the end of the evolutionary search. For GA-I-W, the most selected features were Installation—Repackaging (in 34% of the runs), Installation—Standalone (in 18% of the runs) and Activation—SMS (in 14% of the runs). For GA-ICC-W, the first two most selected features are the same as with GA-I-W: Installation—Repackaging (in 38% of the runs) and Installation—Standalone (in 22% of the runs). However, the third most selected feature by GA-ICC-W is Remote Control—NET (in 14% of the runs).

These features were selected by GA-INFO-W models in the best individuals evolving during the search process but only one feature was selected in each individual (the difference in the selected features is coming from different runs). In order to also test the relative importance and relevance of next features that would be selected, a minimum number of features was set to be selected by

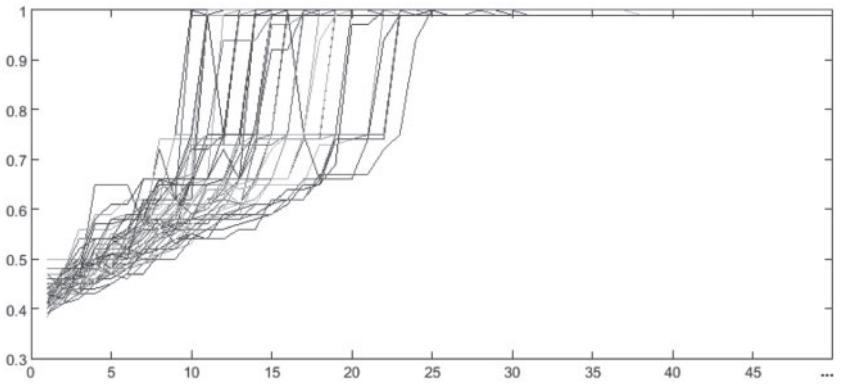

FIG. 4. Fitness values for GA-I-W over each generation in the 50 algorithm runs.

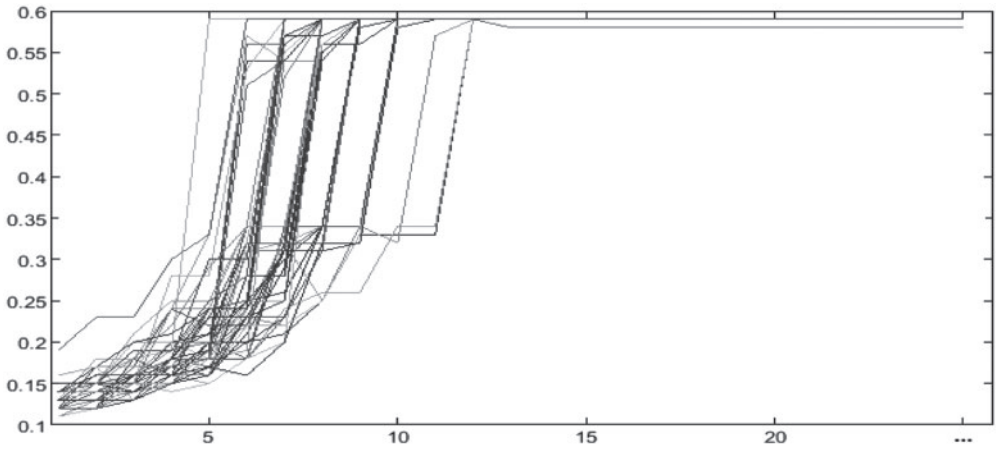

FIG. 5. Fitness values for GA-ICC-W over each generation in the 50 algorithm runs.

both the GA-I-W and GA-ICC-W models. This minimum number of features was set to 2/3/4/5 and the results can be directly contrasted with MRMR selection. Table 2 presents the selection of features made by GA-I-W and GA-ICC-W for minimum number of features $m = 2/3/4/5$. Due to the weighted scheme in the proposed fitness function (see Equation (4)), the number of features selected remained m by the end of the evolutionary search process.

The features selected by the different algorithms are shown in Table 3 and, according to their relevance (% of subsets where it is included), they are ordered in Table 4. While the MRMR method clearly has a low computational cost, it should be noted that the proposed evolutionary approaches are also computationally efficient due to the relatively fast fitness calculation (for both ICC and I measures) and the low population size considered. Moreover, the computational cost of the evolutionary methods can be further improved by significantly reducing the number of generations as it was shown that the optimum values in the population were actually reached very early in the search process (around generation 11 according to Figures 2 and 3).

According to results shown in Table 4, features Installation—Repackaging, Activation—SMS, Remote Control—NET and Financial Charges—SMS have been selected by all the algorithms with

TABLE 2. Selected features by GA-I-W and GA-ICC-W for different values of m (minimum number of features allowed in the subset)

	GA-ICC-W				GA-I-W			
Feature	$m=2$	$m=3$	$m=4$	$m=5$	$m=2$	$m=3$	$m=4$	$m=5$
Installation—Repackaging	√	√	√	√	√	√	√	√
Installation—Standalone	√	√	√	√	√	√	√	√
Activation—SMS			√	√			√	√
Remote control—NET		√	√	√		√	√	√
Financial charges—SMS				√				√

TABLE 3. Selected features by MRMR and GA-INFO-W ($m=5$)

Feature	MRMR	GA-ICC-W	GA-I-W
Installation—Repackaging	√	√	√
Installation—Standalone		√	√
Activation—SMS	√	√	√
Activation—BOOT	√		
Remote control—NET	√	√	√
Financial charges—SMS	√	√	√

TABLE 4. Relevance of selected features

Feature	$m=2$	$m=3$	$m=4$	$m=5$
Installation—Repackaging	100%	100%	100%	100%
Activation—SMS	33%	33%	100%	100%
Remote control—NET		66%	100%	100%
Financial charges—SMS				100%
Installation—Standalone	66%	66%	66%	66%
Activation—BOOT		33%	33%	33%

an associated relevance of 100% (when $m=5$). Hence, it can be concluded that these features are the key ones for the characterization of Android malware families.

From the final selected features above (see Table 4), Installation—Repackaging is the most relevant one as ranked by both MRMR and evolutionary approaches when the search was focused on selecting a minimum-size feature subset with no more than 2 features. The repackaging way of installation was defined by the authors of the dataset [5] as 'one of the most common techniques malware authors use to piggyback malicious payloads into popular applications (or simply apps). In essence, malware authors may locate and download popular apps, disassemble them, enclose malicious payloads, and then re-assemble and submit the new apps to official and/or alternative Android Markets.' Furthermore, from the collected samples, dataset authors found that 1,083 of them (86.0%) were repackaged versions of legitimate applications with malicious payloads.

Regarding the remote control feature, dataset authors stated [5] that 93.0% of the samples turn the infected phones into bots for remote control. Moreover, 1,171 of the samples use the HTTP-based web traffic to receive bot commands. In present experiments, Remote Control—NET was selected by all algorithms for $m=4$ and $m=5$ and in 66% of cases for $m=3$.

In a second order of importance, Installation—Standalone (relevance 66% for all values of m considered) and Activation—SMS (relevance 33% for $m=2$ and $m=3$ and 100% for $m=4$ and $m=5$), together with Activation—BOOT (relevance 33% only for $m=3$, $m=4$ and $m=5$) and Financial Charges—SMS (relevance 100% only for $m=5$) in a third order of importance, have been identified as key features for characterizing malware families.

6 Conclusions and future work

In present paper several methods for selecting those features that best characterize malware families have been proposed. An evolutionary algorithm using a binary representation and a fitness function based on information theory measures (mutual information and ICC) have been developed for the selection of the optimal subset of features in the considered problem. A weighted scheme to balance the relevance of features and the size of the selected subset has also been developed and applied to the well-known Malgenome dataset. Experimental results show that the applied methods agree on the selection of 4 out of the 6 major features, namely: Installation—Repackaging, Activation—SMS, Remote Control—NET and Financial Charges—SMS.

Future work will extend these methods to consider other measures as fitness functions in evolutionary search or other population-based search heuristics. A hybridization of such methods and MRMR-based approaches will also be investigated. Additionally, the applicability of these methods to some other publicly available datasets for the characterization of Android malware families will be further explored.

Acknowledgements

This research has been partially supported through the project of the Spanish Ministry of Economy and Competitiveness RTC-2014-3059-4. The authors would also like to thank the BIO/BU01/15 and the Spanish Ministry of Science and Innovation PID 560300-2009-11.

References

[1] Statista. *The Statistics Portal*. Available from: http://www.statista.com/statistics/266219/global-smartphone-sales-since-1st-quarter-2009-by-operating-system/ (accessed 19 April 2016).

[2] AppBrain Stats. Available from: http://www.appbrain.com/stats/stats-index (accessed 19 April 2016).

[3] F-Secure. *Q1 2014 Mobile Threat Report*, 2015.

[4] *Mind the (Security) Gaps: The 1H 2015 Mobile Threat Landscape*. Available from: http://www.trendmicro.com/vinfo/us/security/news/mobile-safety/mind-the-security-gaps-1h-2015-mobile-threat-landscape (accessed 19 April 2016).

[5] Z. Yajin and J. Xuxian. *Dissecting Android Malware: Characterization and Evolution*. In *2012 IEEE Symposium on Security and Privacy*, pp. 95–109, 2012.

[6] *Malgenome Project*. Available from: http://www.malgenomeproject.org/ (accessed 19 April 2016).

[7] J. Sedano, C. Chira, S. Gonzalez, Á. Herrero, E. Corchado and J. R. Villar. On the selection of key features for Android malware characterization. In *International Joint Conference*, Á. Herrero, B. Baruque, J. Sedano, H. Quintian and E. Corchado, eds, pp. 167–176. Springer International Publishing, 2015.

[8] H. Peng, F. Long and C. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, 1226–1238, 2005.

[9] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, **3**, 185–205, 2005.

[10] S. Arshad, M. A. Shah, A. Khan and M. Ahmed. Android malware detection & protection: a survey. *International Journal of Advanced Computer Science and Applications*, **7**, 463–475, 2016.

[11] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer and Y. Weiss. "Andromaly": a behavioral malware detection framework for Android devices. *Journal of Intelligent Information Systems*, **38**, 161–190, 2011.

[12] L. Gheorghe, B. Marin, G. Gibson, L. Mogosanu, R. Deaconescu, V.-G. Voiculescu and M. Carabas. Smart malware detection on Android. *Security and Communication Networks*, **8**, 4254–4272, 2015.

[13] L. Cen, C. S. Gates, L. Si and N. Li. A probabilistic discriminative model for Android malware detection with decompiled source code. *IEEE Transactions on Dependable and Secure Computing*, **12**, 400–412, 2015.

[14] B. Sanz, I. Santos, C. Laorden, X. Ugarte-pedrero, J. Nieves, P. G. Bringas and G. Alvarez Marañón. MAMA: Manifest analysis for malware detection in Android. *Cybernetics and Systems*, **44**, 469–488, 2013.

[15] S. B. Almin and M. Chatterjee. A novel approach to detect Android malware. *Procedia Computer Science*, **45**, 407–417, 2015.

[16] P. Teufl, M. Ferk, A. Fitzek, D. Hein, S. Kraxberger and C. Orthacker. Malware detection by applying knowledge discovery processes to application metadata on the Android market (Google Play). *Security and Communication Networks*, **9**, 389–419.

[17] O. Somarriba, U. Zurutuza, R. Uribeetxeberria, L. Delosieres and S. Nadjm-Tehrani. Detection and visualization of Android malware behavior. *Journal of Electrical and Computer Engineering*, **2016**, 2016.

[18] J.-W. Jang, J. Yun, A. Mohaisen, J. Woo and H. K. Kim. Detecting and classifying method based on similarity matching of Android malware behavior with profile. *SpringerPlus*, **5**, 1–23, 2016.

[19] A. Feizollah, N. B. Anuar, R. Salleh and A. Wahid. A review on feature selection in mobile malware detection. *Digital Investigation*, **13**, 22–37, 2015.

[20] P. Vinod, V. Laxmi, M. S. Gaur and S. Naval. MCF: MultiComponent features for malware analysis. In *27th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 1076–1081, 2013.

[21] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, **3**, 1157–1182, 2003.

[22] P. Larrañaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armañanzas, G. Santafé, A. Pérez and V. Robles. Machine learning in bioinformatics. *Briefings in Bioinformatics*, **7**, 86–112, 2006.

[23] H. Liu, L. Liu and H. Zhang. Ensemble gene selection by grouping for microarray data classification. *Journal of Biomedical Informatics*, **43**, 81–87, 2010.
[24] Y. Saeys, I. Inza and P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, **23**, 2507–2517, 2007.
[25] N. Hatami and C. Chira. Diverse accurate feature selection for microarray cancer diagnosis. *Intelligent Data Analysis*, **17**, 697–716, 2013.
[26] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

Received 3 March 2016