SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks (2011)
DOI: 10.1002/sec.341

SPECIAL ISSUE PAPER

Evaluation of anomaly-based IDS for mobile devices using machine learning classifiers

Dimitrios Damopoulos[1]*, Sofia A. Menesidou[1], Georgios Kambourakis[1], Maria Papadaki[2], Nathan Clarke[2] and Stefanos Gritzalis[1]

[1] Info-Sec-Lab Laboratory of Information and Communications Systems Security, University of the Aegean, Samos, Greece
[2] Centre for Security, Communications and Network Research University of Plymouth, Plymouth, U.K.

ABSTRACT

Mobile devices have evolved and experienced an immense popularity over the last few years. This growth however has exposed mobile devices to an increasing number of security threats. Despite the variety of peripheral protection mechanisms described in the literature, authentication and access control cannot provide integral protection against intrusions. Thus, a need for more intelligent and sophisticated security controls such as intrusion detection systems (IDSs) is necessary. Whilst much work has been devoted to mobile device IDSs, research on anomaly-based or behaviour-based IDS for such devices has been limited leaving several problems unsolved. Motivated by this fact, in this paper, we focus on anomaly-based IDS for modern mobile devices. A dataset consisting of iPhone users data logs has been created, and various classification and validation methods have been evaluated to assess their effectiveness in detecting misuses. Specifically, the experimental procedure includes and cross-evaluates four machine learning algorithms (i.e. Bayesian networks, radial basis function, *K*-nearest neighbours and random Forest), which classify the behaviour of the end-user in terms of telephone calls, SMS and Web browsing history. In order to detect illegitimate use of service by a potential malware or a thief, the experimental procedure examines the aforementioned services independently as well as in combination in a multimodal fashion. The results are very promising showing the ability of at least one classifier to detect intrusions with a high true positive rate of 99.8%.

KEYWORDS
mobile devices; anomaly-based intrusion detection system; user behaviour; machine learning classifiers

***Correspondence**
Dimitrios Damopoulos, Laboratory of Information and Communications Systems Security, Department of Information and Communication Systems Engineering, University of the Aegean, Karlovassi, Samos GR-83200, Greece.
E-mail: ddamop@aegean.gr

1. INTRODUCTION

Mobile devices have evolved and experienced a great success over the last few years [1]. Such devices are capable of performing sophisticated tasks and communicate through various wireless interfaces [2]. However, along with their popularity, mobile devices face an everyday growing number of security threats [3,4]. This is despite the variety of peripheral protection mechanisms proposed in the literature in recent years. Without doubt, authentication and access control methods can be used in many cases, but alone, they are not sufficient to offer integral protection against intrusions. Overall, with the increasing risk of mobile malware, the theft or loss of mobile devices and the physical vulnerability, that is, rewiring a circuit on a chip or using probing pins to monitor data flows to retrieve private keys or find flaws in the hardware components [5], designing a highly secure mobile device is still a very challenging task.

While more than four billion people [6] enjoy their mobile devices using 2G/3G mobile networks, Kaspersky Lab has very recently identified 39 new mobile malware families (SMS trojans, iPhone malware, Android spyware) with 143 modifications [7]. According to a ScanSafe report, malware volumes grew 300% in 2008, and it is noted that most of the legitimate web pages crawling on the Internet are not trustworthy or infected by different kinds of viruses [8]. Moreover, according to the UK Home Office, 69% of robberies include a mobile device [9]. As a result, a need for more intelligent and sophisticated security controls such as intrusion detection systems (IDSs) for mobile devices is necessary. In general, there are two basic approaches in IDS to detect an intrusion: (i) *misuse based* (also called signature based), and (ii)*anomaly based* (also called behaviour based). Although misuse-based IDS can immediately be employed to monitor the mobile environment, only an anomaly-based IDS is able to detect new, unforeseen vulnerabilities and variants of known attacks. Anomaly-based intrusion detection profiles normal behaviour and attempts to identify patterns of user activities that deviate from a predefined or dynamically updated profile [10]. Although much research has been devoted to IDS, in the context of anomaly detection, the exploration of what is defined as 'normal' has been limited, and several important problems remain unsolved [11].

In this paper, we concentrate on anomaly-based IDS for modern mobile devices. After gathering a significant number of iPhone users' data (profiles), we created our own input dataset for the experimental detection process. The goal here is to detect anomalies, that is, actions that deviate from the normal behaviour of the legitimate user. Of course, as already pointed out, such actions may arise for a number of reasons, including malware, illegal use of the device and so forth. Specifically, every user profile gathered directly from the mobile device includes all logs from telephone calls, SMS and Web browsing services. The collection process has been fostered by a client–server solution and special care has been taken to preserve the participants' privacy and anonymity. Four different machine learning classifiers have been thoroughly examined, that is, Bayesian networks, radial basis function (RBF), *K*-nearest neighbours (KNN) and random forest based on their performance, speed and ability to detect anomalies. Also, our experiments take into account two different types of well known validation methods, namely 66% split and 10-fold cross-validation.

The data analysis has been performed considering a proxy-assisted IDS system, whereas the implementation of the corresponding host-based IDS is left for future work. Our findings show the ability of random forest to successfully detect misuse of Telephone call, SMS and Web browsing services by achieving a 99.8% true positive rate (TPR) (also referred to as sensitivity) and contributing about 1.2% of TPR from previous researches. Moreover, for the average error rate ($1 - \text{accuracy}$) and false negative rate ($FNR = 1 - TPR$), we obtained remain less than 1.6% and 0.7%, respectively. We extensively discuss our findings that aside from TPR, consider other important metrics like accuracy, response time and receiver operating characteristic (ROC) curve analysis. Note that to the best of our knowledge, this is the first work that attempts to classify intrusions using four popular machine learning algorithms and takes into consideration the Web browsing service as well. Another important contribution of this work is that we examine the telephone call, SMS and Web browsing services logs not only separately but also combined in a multimodal fashion.

The rest of the paper is organised as follows. The next section addresses previous work on the topic. Section 3 presents the methodology used throughout this work. First, some issues about the data collection process are discussed. Second, it provides a description of the work carried out to extract knowledge from the collected data, the statistical analysis and the classification experiments. Section 4 presents and discusses the evaluation outcomes of this work. Section 5 provides single-user ROC curves experiments [12], which are used to identify the quality of a possible mobile device IDS using the aforementioned algorithms. Finally, Section 6 concludes the paper and provides some future directions.

2. RELATED WORK

The work in [13] proposed a prototype of a tool, based on a supervised artificial neural network (ANN), to detect anomalous behaviour on mobile communications, such as service fraud and subscriber identity module card cloning. The authors, based on their prototype, report accuracy of a 92.50% detection of fraudulent users and a 92.5% correct classification of legitimate users. The work in [14] proposed the Bayes decision rule towards the generation of mobility user profiles within the global system for mobile communications (GSM) network. By utilising their method, the authors managed to achieve a TPR of 83.50%. One problem with this approach is the privacy of the end-user's usage log files, which are exposed to the telecom carriers in order to detect mistrusted users, as explained in [15].

Hollmén [16] has proposed fraud detection techniques in mobile networks by means of user profiling and classification. Specifically, the author used ANN and probabilistic models to detect anomalous usage and achieved a TPR of 69%. However, the presented method for fraud detection is based on an available large database with billions of records. As a result, this method can be seen only as a specific user profiling problem in fraud detection. The authors in [17] used ANN to form short and long-term statistical behaviour profiles for GSM and universal mobile telecommunication systems networks. They define two time spans over the call data records, that is, a shorter sequence or current behaviour profile and a longer one or behaviour profile history. They also used the maximal entropy principle to create statistical profiles and the Hellinger distance to calculate the distance between current behaviour profile and behaviour profile history. If this distance is greater than some pre-determined threshold, an alarm is raised.

The authors in [15] discussed how ANN and other tools can be applied against frauds in first generation (1G) mobile networks. They also presented an online security system for fraud detection of mobile phone operations using the RBF model. They have pointed out that it is very hard to build a system capable of identifying any possible fraud; however, they managed a TPR of 97.50%. Also, the authors in [18] proposed an online anomaly detection algorithm, based on Markov Model, where the key distinguishing characteristic is the use of sequences of network cell IDs traversed by a user. With this IDS, they attempted to address the problem of subscriber identity module cloning and media access control address spoofing. Through their experimental procedure, a TPR of 87.50% has been attained. The work in [3] proposed a mobility-based anomaly detection scheme to detect cloning attacks and cell phone losses. .The authors employed several methods, such as high order Markov techniques, the exponentially weighted moving average model and the Shannon's entropy, in order to explore normal usage profile. The highest TPR they achieved was 89%.

Recently, in [19] the authors presented a test bed for experimenting with anomaly detection algorithms and demonstrated its properties using two unsupervised anomaly detection methods, that is, self-organizing map and clustering. They concluded that both methods are suitable for network monitoring. The work in [20] presented a behavioural detection framework for malware targeting mobile devices. Particularly, the framework generates a malicious behaviour signature database by extracting the key behaviour signatures from the mobile malware. By using this scheme, the authors tried to apply a method called temporal logic of causal knowledge in order to address the challenge of behavioural detection. This is managed by providing a compact 'spatial-temporal' representation of program behaviour. To identify malicious behaviour, they used support vector machine (SVM) classification to train a classifier from both normal and malicious data. Their evaluation on both simulated and real-world malware samples indicates that behavioural detection is able to identify current mobile viruses and worms with more than 96% accuracy. The authors in [21] proposed VirusMeter, a malware detection system and cross-evaluated linear regression, ANN and decision trees, for their ability to detect anomalous behaviours on mobile devices. By monitoring power consumption on a mobile device and using ANN, they achieved a TPR of 98.60%. However, VirusMeter detection can be affected because the precision of battery power indicators may vary significantly between different mobile operating systems (OS).

Table I summarises all methods used including TPR and FPR (where it is referred) for the aforementioned anomaly-based mobile IDSs. We can easily observe that the most frequently applied technique is the ANN [13,15–17,19,21]. Note that ANN is a sophisticated technique capable of modelling extremely complex functions [22]. Some other classification techniques that have been employed so far are knowledge-based [13], Bayes decision rule [14], probabilistic methods [16], Markov model [3,18], SVM [20], linear regression and decision trees [21]. Finally, clustering techniques have been used for mobile network intrusion detection as well [19].

Despite the fact that all the aforementioned researches have significantly contributed to the anomaly-based IDS for mobile devices issued, several important problems remain unsolved. Currently, the main disadvantage of most IDS for mobile devices that use anomaly detection techniques is the high false alarm rate (FPR) [23]. Hence, there is an urgent need for methods that substantially improve the detection rate while minimising false alarms. Also, so far, the literature focused only on cellular networks and in particular in telephony and SMS services. Nevertheless, mobile devices have evolved and experienced a great success over the last few years being capable of performing sophisticated tasks and communicate through various interfaces [2]. Thus, it is very important for any analysis of user profiles to take into account the data originating from the provision of other services such as Web browsing, email and so forth. This way the IDS would be more effective in detecting abnormalities in behaviour, which naturally may be induced not only by malicious individuals but also by stealth malware running on the mobile device. For example, this may happen when the malware tries to send a considerable amount of intercepted information via SMS and/or use telephone numbers that have not been dialled by the legitimate user in the past [24].

Table I. Method and corresponding TPR for anomaly-based mobile IDS.

Year	Reference	Technique	TPR (%)	FPR (%)
1997	Moreau *et al.* [13]	Supervised and unsupervised ANN	N/A	N/A
1998	Buschkes *et al.* [14]	BDR	83.50	N/A
2000	Hollmen [16]	ANN/probabilistic methods	69.00	16.0
2001	Burge and Shawe-Tylor [17]	Unsupervised ANN	N/A	N/A
2002	Boukerche and Notare [15]	ANN (RBF)	97.50	N/A
2004	Sun *et al.* [18]	Markov model	87.50	3.0
2006	Sun *et al.* [3]	Markov model	89.00	5.0
2008	Kumpulainen and Hatonen [19]	ANN (SOM)/Clustering	N/A	N/A
2008	Bose *et al.* [20]	SVM	N/A	6.6
2009	Liu *et al.* [21]	ANN	98.60	4.3

ANN, artificial neural network; BDR, Bayes decision rule; FPR, false alarm rate; RBF, radial basis function; SOM, self-organizing map; SVM, support vector machine; TPR, true positive rate.

3. METHODOLOGY

Taking into account the above discussion, we concluded that there is a need for more intelligent and sophisticated security controls, such as anomaly-based IDSs, to tackle mobile device intrusions. To do so, various user's actions or behaviours performed on the mobile device should be collected in order to create behavioural profiles and to effectively discriminate legitimate users from intruders. Several features of the collected dataset can be used as input for a number of machine learning classifiers to investigate and optimise the performance of an anomaly-based IDS. In this stage, data analysis can be performed offline or assigned to a proxy server. Later on, by capitalising on the results, one can build a dynamically updated host-based IDS that runs directly on the mobile device in real-time. In this section, we provide information on the data collection process, the type and structure of data that we are going to analyse, as well as on the selected validation and classification methods.

3.1. Data collection

Earlier research in the field of mobile IDS has particularly focused on telephone calls and SMS in order to detect illicit use of services. Nevertheless, as already pointed out, nowadays, users do not employ their mobile device only for these basic services, but they also use it for a variety of other services such as Web browsing. For this reason, our research is not only bound to collect data from telephone calls and SMSs, but also the Web browsing history ones. With the variety of these data, we attempted to create an integrated user behaviour profile that combines the most popular services and hence can better depict user's normal behaviour.

The main problem of the data collection process is to find a critical mass of users (sample) that are willing to provide us with their sensitive data for the need of this research. Even though the data will be collected in an anonymised form, it is very difficult for someone to supply them. Also, the plethora of different mobile devices and OSs makes the collection of such private data more difficult. Specifically, each mobile OS stores these data differently. In addition, most of the modern mobile OSs keep user-sensitive files or databases along with kernel's data. Therefore, because of the sensitivity of such data, the access privileges are limited in the general case. Indeed, all the latest mobile OSs do not allow access to these files in order to protect the privacy of the end-users. In some cases, the only way to gain access to this information is to somehow bypass root privileges. However, this raises ethical problems and at the same time reduces the number of willing-to-participate individuals in such a research. Last but not the least, to facilitate such a research it is necessary to provide a straightforward data collection method.

To cope with the aforementioned problems, we decided to collect data from only one popular and modern mobile device. iPhone (Apple Inc., Cupertino, CA, USA) is a modern worldwide mobile device with over 50 million items sold until April 2010 [25]. Moreover, iPhone, like any other ultramodern mobile device, supports a variety of different (mainly wireless) network technologies. Through these network interfaces, mobile devices are able to synchronise with desktop computers. iPhone's iOS is not only able to synchronise with a desktop computer but at the same time can automatically keep backup files in the same machine. These backup files are kept in structured query language databases and in property lists files. Therefore, iPhone backup is the proper solution to the data collection issue.

In order to collect the required data and simplify the data collection process, the iBackup client–server system has been created. The iBackup server is hosted within the University's domain and consists of a Web site (http://ibackup.samos.aegean.gr/), a database and the iBackup server application. Every iPhone user is able to participate in the data collection process, by simply downloading the iBackup client. This client is the main application that is used to facilitate the data collection process. Because iPhone is able to synchronise [26] only with Windows and Macintosh OSs, the data can be collected only through these OSs. Table II summarises the iPhone files required for each particular service and the particular features that we choose to collect and use in the experiments. The only collected properties from which user's information can be leaked are the telephone numbers and the Web site hyperlinks that the user has visited. Hence, in order to preserve user's anonymity, a hash function, namely SHA-1 has been used [27]. By doing so, unlinkability is preserved because there is no such a way to link user's true data with specific published data in the server side. A detailed analysis of these properties is given in the next section.

3.2. Data structure

According to our study four scenarios of experiments have been conducted for all the users in the sample. The first three of them focus on telephone call, SMS and Web browsing services having each service analysed indepen-

Table II. Collected data and their features.

Collected data	Corresponding iPhone file	Collected features
Telephone calls	call_history.db	Number, timestamp, flag (incoming or outgoing), duration
SMS	sms.db	Number, timestamp, flag (incoming or outgoing), country
Web browsing history	History.plist	Web site link and timestamp

dently. Specifically, as it is discussed in the following, for each particular service, N data files have been created where a vector of associated features has been stored per event. Hence, each file contains the data of the corresponding legitimate user and the data of the rest $N-1$ users that represent the potential intruders. This means that, for each user in the dataset, the corresponding data file contains the following: (i) the user's personal data, referred to as normal behaviour data, and (ii) all other users' data that represent potential illegal behaviours.

Every record of the telephone call data file is composed of the following collected features. First, the feature *Number* refers to the telephone number of the caller or the callee. This field has been anonymised via the use of the SHA-1 hash function. The *Timestamp* feature refers to aUNIX timestamp (based on seconds since standard epoch of 1/1/1970) and represents the date and time a telephone call took place. Next, the *Flags* feature indicates the direction of a call, that is, incoming or outgoing. The *Duration* feature represents the duration of a call in seconds. Last, the *Intruder* feature is binary representing the two nominal classes, that is, if this data belong to the legitimate user (no) or the potential intruder (yes). An example of such a record is given by the following quintuplet (vector) {7e738835c130ec478ec8ae99707a4a5eeabd25c6, 1252676780, 60, 0, no}

Each record of the SMS data file in turn is composed of the following features. The *Number* feature refers to the mobile number that the particular message has been sent or received. This feature has been anonymised as well. The *Timestamp* feature represents a UNIX date and is referred to the date and time that an SMS has been sent or received. Next, the *Flags* feature indicates the direction of an SMS (incoming or outgoing). The *Country* feature represents the country of the sender or the receiver. Last, the *Intruder* property is binary representing the two nominal classes, i. e. the legitimate user (no) or an intruder (yes).

The records of the Web browsing history data file are composed of three features. The *Web site Link* feature, which is anonymised, refers to the visited web site. Next, the *Timestamp* feature corresponds to the date and time that the Web site has been accessed. Last, the *Intruder* feature represents the two nominal classes, a legitimate user (no) or the intruder (yes).

According to the last scenario, we create a multimodal that integrates the evidence presented by multiple services. Specifically, this scenario is a fusion of telephone call, SMS and Web browsing service data. In this way, we represent the behaviour of the end-user as discrete events, which take place at a specific timestamp. To realise this blend of information, data files have been created where a set of relevant features have been stored for each one of the N users. As with the first three scenarios, each data file is represented by only one legitimate user, and the rest $N-1$ users behave as potential intruders. The multimodal data files are composed of the *Event*, *Timestamp* and *Intruder* features. All the three features correspond to information similar to what was described in the previous paragraph. An example of the Multimodal data file is given by the following triplet (vector) {1b4766fca21995aa15f2-bed0d25db5014e73ab94, 1257843913, yes}.

3.3. Methods

To predict and classify potentially unauthorised actions and malicious occurrences in user behaviour, while minimising the rate of incorrect flagging, various machine learning classifiers have been utilised. Specifically, the analysis procedure takes into account and cross-evaluates four supervised machine learning algorithms, that is, Bayesian networks, RBF, KNN and random forest, which pattern the behaviour of the end-user, in terms of telephone calls, SMS, Web browsing history and multimodal information. A Bayesian network, also called a belief network model, is an annotated directed graph that encodes the probabilistic relationships among variables of interest. A Bayesian network classifier is a statistical classification eager method [28] that maybe used as a classifier that gives the posterior probability distribution of the class node given the values of other features. On the other hand, RBF is a type of ANN that consists of an input layer, a hidden layer and an output layer. Specifically, RBF is a single hidden layer feed-forward network and has a static Gaussian function as the nonlinearity for the hidden layer processing elements [29]. KNN is one of the simplest classification methods so far. Also, KNN is a type of instance-based learning, also known as lazy learning classification and is based on the Euclidean distance. A KNN algorithm should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data [30]. Last, the random forest is an ensemble of decision trees such that each tree depends on the values of a random vector. This vector is sampled with the same distribution for all trees in the forest and is totally independent. Random forest is well-respected amongst the statistics and the machine learning communities as a versatile eager method that produces accurate classifiers for many types of data [31].

For all the scenarios, two different types of well-known validation methods have been employed to divide the dataset into different sub-samples. The first one is a percentage split and more specifically a 66% split validation. The holdout or percentage split method splits the dataset randomly into two groups, called the training set and the testing set. The training set (66%) is used to train the classifier, whereas the test set (the rest 34%) is used to estimate the error rate of the trained classifier. The second method is a k-fold cross-validation and more specifically a 10-fold one. A k-fold cross-validation method is a way to improve the holdout method. The original sample is randomly divided into k equally (or nearly equally) sized sub-samples, and the cross-validation process is repeated k times (the folds). Each time, one of the k sub-samples is used as the test set and the other $k-1$ sub-samples are put together to form the training set. Finally, the average error across all k trials is computed [32].

The totally different way these methods operate will help us to better estimate their impact in the final classification results in terms of accuracy and speed. Thus, although the 66% split method is expected to be faster than the 10-fold cross-validation, it is unclear if and how much this might affect the classification results.

The analysis of the collected data has been performed on a laptop machine with an Intel Core 2 Duo T7200 CPU (Intel Corp., Santa Clara, CA, USA) at 2 GHz and 3.2 GB of RAM. The OS of this machine is Microsoft Windows 7 (Microsoft Corp., Redmond, WA, USA). Also, data analysis was carried out using the well-known machine learning software package namely Waikato Environment for Knowledge Analysis (Weka) [33] with 1 GB memory as the upper bound to carry out the final classification experiments. The Java Runtime Environment in version 1.6.0_17 has been used for Weka parameterisation according to the guidelines provided in [34–37].

Moreover, in order to select the most appropriate machine learning algorithms to be used throughout the data analysis, some preliminary classification tests among common machine learning classifiers have been conducted in terms of service time and memory consumption. Bear in mind that this is a very important task in order to eventually select those algorithms that are more suitable for a host-based IDS. Two types of initial tests have been carried out to address the mobile device memory limitations: the first one with an upper bound of 128 MB and the second with 2 GB of memory. We selected these bounds because they correspond to the typical read-only memory and storage used in modern mobile devices. Also, for these tests, an amount of telephone calls, SMS, Web browsing history and multimodal data files have been chosen randomly. The results showed that multilayer perceptron is not able to run when low to moderate memory usage, that is, ≤2 GB is selected. Moreover, SVM runs only with an upper bound of 2 GB and only if using telephone calls, SMS and Web browsing history data files. For the aforementioned reasons, we decided to exclude these two algorithms from our experiments. It is also stressed that random forest—due to these memory limitation—was not able to create the necessary decision trees in order to classify the Multimodal data files. Table III summarises all the employed classifiers as well as the obtained results in every case. Overall, this study left us with the first four algorithms for further evaluation.

Table III. Preliminary classification tests.

Algorithm	128 MB	2048 MB	Time (s)
Bayesian networks	T, S, W, M	T, S, W, M	0–3
RBF	T, S, W, M	T, S, W, M	0–26
KNN	T, S, W, M	T, S, W, M	0–131
Random forest	T, S, W	T, S, W	0–7
SVM	*	T, S, W	>3600
MLP	*	*	*

KNN, *K*-nearest neighbours; MLP, multilayer perceptron; RBF, radial basis function; SVM, support vector machine.

T, telephone calls; S, SMS; W, Web browsing history; M, multimodal; *, will not run.

4. RESULTS

In this section, we cross-evaluate four machine learning classifiers in terms of performance and effectiveness to detect intrusions. Also, we consider two different types of validation methods to estimate their effect in the final results. This assessment is not only necessary to ideally find out the best classifier but also to end up to those that are more suitable for a host-based IDS (i.e. the ones that can run directly on the mobile device).

4.1. Descriptive facts

The dataset is consisted of 35 iPhone users' data, and the participants came from two different countries, Greece and the UK. The collected data consist of 8297 telephone calls, 11 321 SMSs and 790 hyperlinks. Figure 1 is a snapshot of all the participants' behaviour profiles and depicts characteristically the uniqueness of mobile usage per user. As expected, all the participants use their mobile devices to make telephone calls and exchange SMSs. On the other hand, about 66% of the subjects use their mobile device to access the Internet.

Also, when analysing the user's mobile profiles, we note that only a small percentage of their behaviour is unique. For example, only 2% of the SMSs have been sent to or received from unique mobile numbers. This means that 98% of SMSs has been sent to or received from the same user at least twice. The same behaviour is observed for Telephone calls and Web site visiting, having a percentage of 3% and 9%, respectively.

4.2. Effectiveness

We consider two metrics to estimate the effectiveness of the IDS: first, the TPR, which is the probability of an alarm given an actual intrusion, and second, the accuracy, which is defined as the sum of true positives and true negatives divided by the total number of events. For both metrics, we consider an average value obtained by taking the statistical average of the values resulting from 35 experiments (i.e. the total number of cases). In majority of the experiments, the TPR metric gave an average value of 99.3%, whereas the average accuracy had a value of 98.5%. As a consequence, the average error rate, which is defined as the incorrectly classified instances, is less than 1.6%, and the average FNR, which is the probability of no alarm given an actual intrusion, is less than 0.7%. Figures 2 and 3 summarise the average TPR and accuracy metrics logged for each sub-scenario, that is, telephone calls, SMSs, Web browsing history and multimodal using the Bayesian networks, RBF, KNN and random forest classifiers. Recall that for each algorithm, we tested two different validation methods, namely 10-fold cross-validation and 66% split.

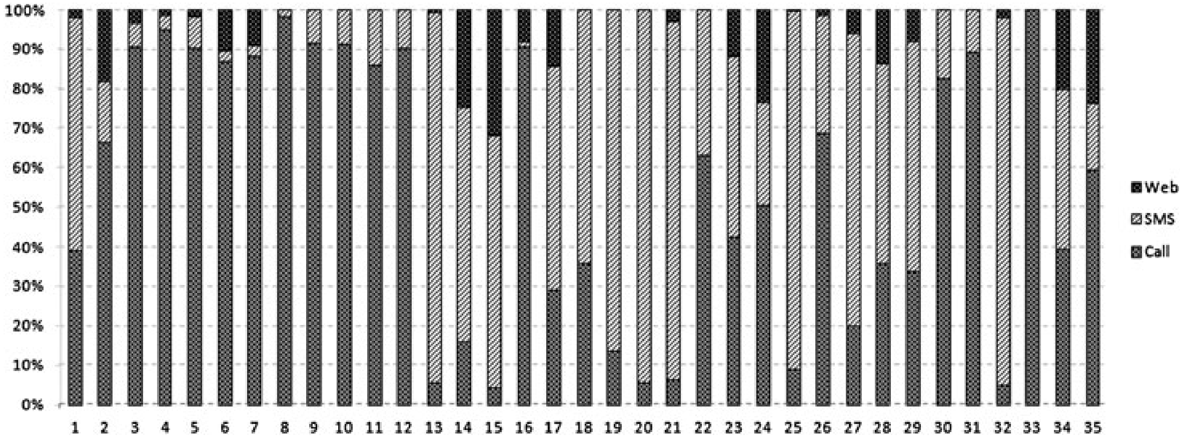

Figure 1. A snapshot of participants behaviour profile.

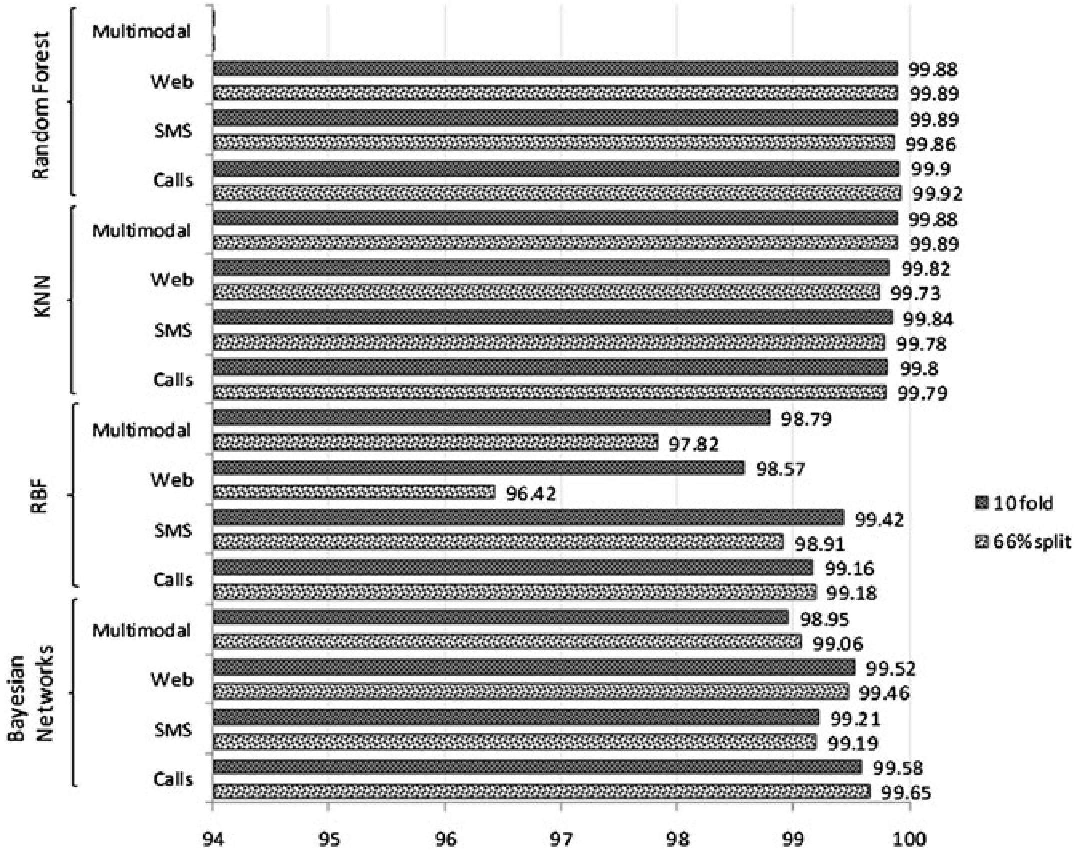

Figure 2. Average TPR (%) per validation method for each algorithm and sub-scenario. KNN, *K*-nearest neighbours; RBF, radial basis function.

Considering the results obtained from telephone calls, SMSs and Web browsing history as separate services, we conclude that random forest is the most promising classifier showing optimal results. Specifically, its average TPR and accuracy remain in all cases above 99.8% and 98.9%, respectively. Note that this observation stands for both validation methods. Bayesian networks and KNN also obtained very promising results, that is, an average TPR and accuracy of 99.06%/98.76% and 99.73%/99.49% in the worst case, respectively. Moreover, in the first three sub-scenarios, KNN scored higher in accuracy, compared with random forest, ≈99.5% vs 99.25% in the worst case, respectively. On the other hand, RBF had the minimum TPR (≈96.4%) and accuracy (≈94.5%). In nearly all cases, the worst accuracy is perceived when analysing the Web history data. This happens because the volume of the

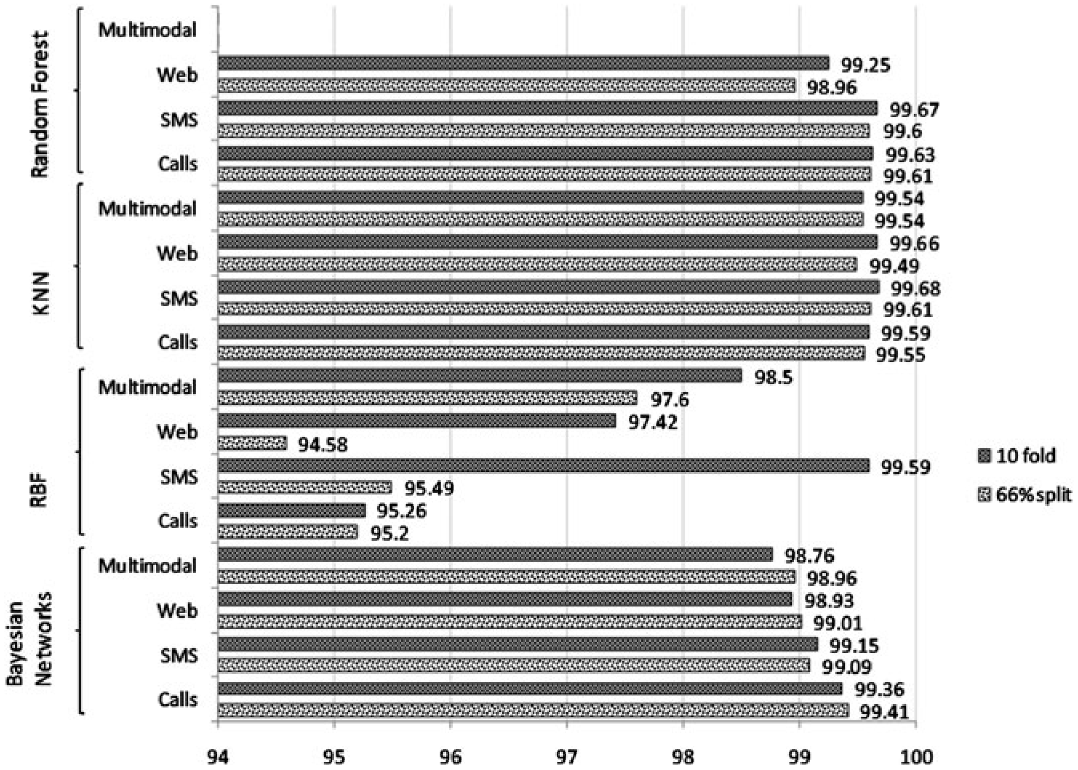

Figure 3. Average accuracy (%) per validation method for each algorithm and sub-scenario. KNN, *K*-nearest neighbours; RBF, radial basis function.

available information collected is less for all users. Naturally, this is expected for the majority of mobile users. Also, it is worth mentioning that the FNR in majority of the experiments remains below 0.7%.

As already pointed out, to optimise the results, we created a multimodal comprising a fusion of telephone calls, SMSs and Web browsing history data. In Multimodal, the best results logged by KNN achieved a TRP and accuracy of 99.80% in the three first experiments. As a general remark, in majority of the experiments, the 10-fold cross-validation method showed 1% better results in contrast to the 66% split validation one.

4.3. Performance

Although, random forest, KNN and Bayesian networks showed very good detection rates, TPR and accuracy are not the only metrics to cross-evaluate the classifiers and shape a better opinion about their efficiency. Another important metric may be the time the IDS needs to reach a decision. This is quite important in mobile devices, which as a general rule, do not afford unlimited computational and memory resources. In this point of view, an algorithm that is able to identify and classify the potential intruders in a small time period is highly appreciated. In this context, we evaluated all classifiers in terms of speed, that is, how much time they need to come to a decision (classification). This is tested both for every type of collected data (information) as well as the two available validation methods.

Table IV offers an aggregated comparative view of the average classification time in seconds for all the scenarios. This is actually the average time needed for each algorithm to classify and verify the results with the testing dataset.

Table IV. Average classification times (in seconds).

	Bayesian networks		RBF		KNN		Random forest	
	10-fold	66% split	10-fold	66% split	10-fold	66% split	10-fold	66% split
Calls	0.9	0.5	4.9	0.7	6.8	1.5	5.5	1.0
SMS	0.7	0.1	8.3	1.6	12.0	3.3	6.0	1.5
Web	<0.1	<0.1	0.7	0.1	<0.1	<0.1	<0.1	<0.1
Multimodal	1.6	0.6	19.6	2.0	77.3	16.0	–	–

KNN, *K*-nearest neighbours; RBF, radial basis function.

Table V. Average classification times in terms of validation methods (in seconds).

	10-fold	66% split
Bayesian networks	0.8	0.3
RBF	8.3	1.1
KNN	24.0	5.2
Random forest	3.8	0.8

KNN, *K*-nearest neighbours; RBF, radial basis function.

We observed that all the algorithms using the 66% split validation method achieve very good performance and classify an intrusion under 3.5 s in all cases. Contrariwise, the 10-fold cross-validation method increases significantly the corresponding times. This is because the 66% split validation procedure is executed once and precedes that of classification, whereas in the 10-fold cross-validation case, these phases are executed 10 times consecutively. In the Web browsing history case, this time was equal to nearly zero computational time for all the employed classifiers. Naturally, this result depends on the volume of data to be analysed, which in this case is limited. Bayesian networks, which is the third best algorithm considering the TPR metric, is also the quickest algorithm here succeeding less than 1 s in all cases but one. KNN and random forest incur a greater penalisation in order to achieve a better classification as already explained. Note that in the multimodal case, Bayesian networks succeed the optimal time to classify correctly an intrusion sacrificing only an average of 0.8% TPR compared with the multimodal KNN case. Also, it is worth mentioning that even though KNN is the sole classifier that improved its performance in the multimodal scenarios, it was the one with the highest delay as well.

Table V offers time comparisons between the two validation methods for all algorithms. As already pointed out, in majority of the experiments, the 10-fold cross-validation method produces a little better result compared with the 66% split one. However, as shown in Table V, using the latter method, the classification procedure has been conducted faster by all algorithms.

5. SINGLE-USER RECEIVER OPERATING CHARACTERISTIC CURVE EXPERIMENT

Receiver operating characteristic curve analysis has been increasingly used in machine learning and data mining to investigate the relationship between sensitivity (TPR) and specificity (1 – FPR) of a binary classifier [38]. An ROC curve represents the trade-off between the percentage of similar shapes correctly identified as similar (TPR) and the percentage of dissimilar shapes wrongfully identified as similar (FPR). Any increase in sensitivity will be accompanied by a decrease in specificity (1, 1). The best performance is provided by curves that pass beside the upper left region (point (0, 1)). This means that the examined IDS provides high detection accuracy with low FPRs. Putting it another way, this point represents 100% sensitivity (no false negatives) and 100% specificity (no false positives), which is also called a perfect classification. The lower left and upper right points correspond to no detection at all [39]. Thus, in the following, we use ROC graphs to further discuss and analyse the most important results given in Section 4.

For ROC analysis, the data of the 13th user has been selected. This user dataset consists of 100 telephone calls, 1698 SMS and 13 Web browsing history entries. Considering the current sample, this distribution of entries per service corresponds to the average user. Figure 4 depicts the obtained ROC curve for this user when utilising the random forest algorithm. This is because random forest scored higher in all scenarios except the multimodal one in terms of TPR and accuracy. The graphs have been derived from the 10-fold cross-validation method for telephone calls (left) and SMS (right) experiments. In the figures, the TPR metric is plotted against that of FPR. We easily noted that both ROC curves are lying in the top left, above the diagonal connecting the lower left and upper right points. Note that the exact coordinates of all the indicated points that appear on the ROC curves correspond to any possible threshold value that the IDS can set to operate.

Figure 5 depicts Web browsing history ROC curves for Bayesian networks (upper left), RBF (upper right), KNN (lower left) and Random Forest (lower right). All graphs

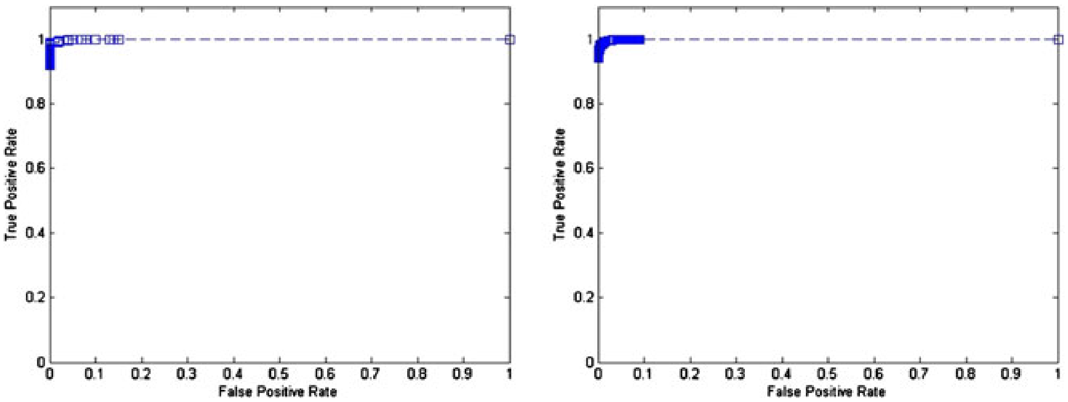

Figure 4. Random forest receiver operating characteristic curves for telephone call and SMS.

have been derived from the 10-fold cross-validation experiments. This time, the results for all the algorithms seem to degrade, but still, all curves tend to lie in the top left corner. For instance, when comparing the plots of Figure 4 with that of Figure 5, we can infer that although random forest presents a good detection rate in the general case, its specificity has been diminished when taking into consideration the Web history data independently. As already pointed out, this penalisation is due to the limited amount of Web browsing data entries, that is, only 13 records in total.

Figure 6 depicts the ROC curves for the multimodal scenario using the KNN algorithm. The graphs have been created when selecting the 10-fold cross-validation (left) and 66% split validation method (right). We easily confirm that KNN improved its specificity in Multimodal as already discussed in Section 4. Last but not the least, when comparing the lower left plot of Figure 5 with those of Figure 6, it is obvious that the multimodal, which blends the user data stemming from all three services, decreases significantly the FPR and achieves almost perfect classification.

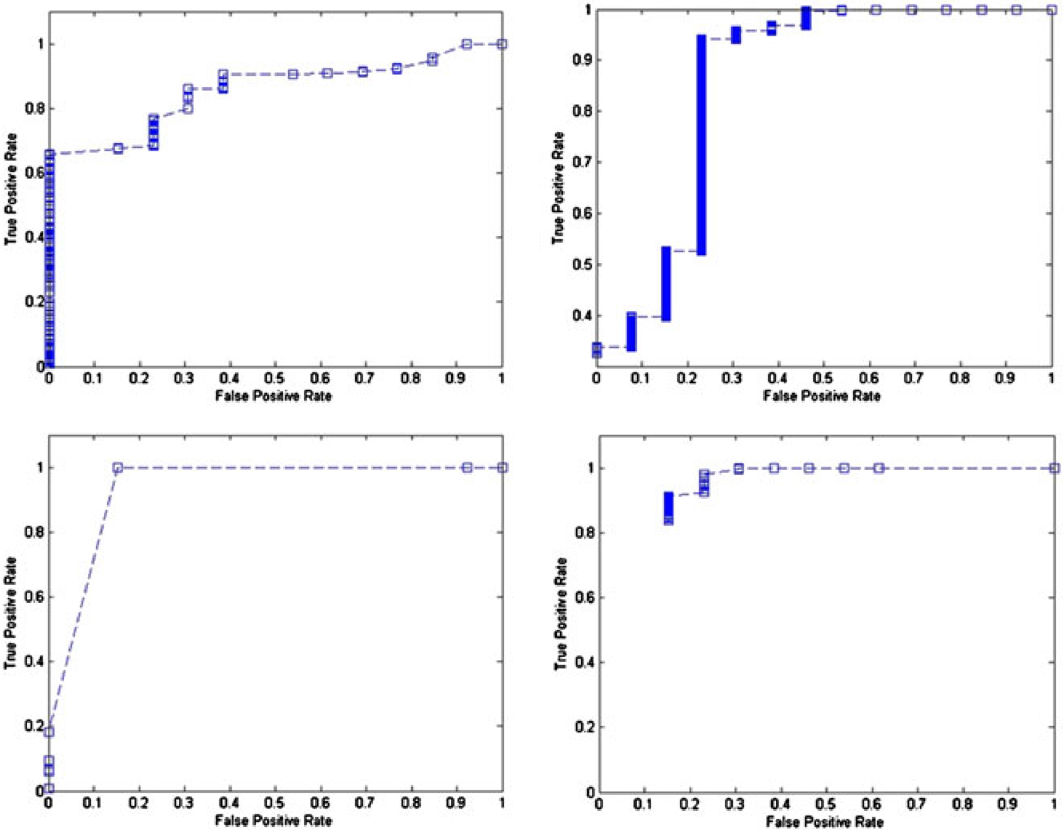

Figure 5. Web browsing history receiver operating characteristic curves (10-fold cross-validation method).

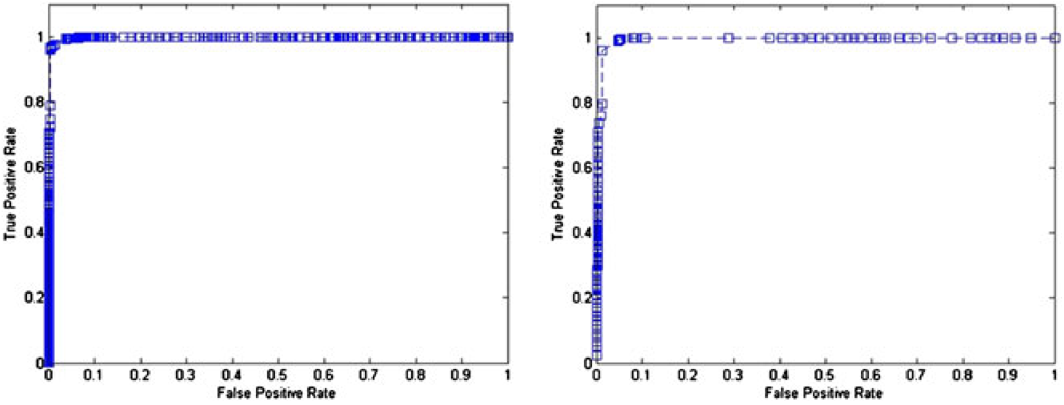

Figure 6. KNN multimodal receiver operating characteristic curves (10-fold cross and 66% split validation methods).

As already mentioned, to find out a fair trade-off between effectiveness and performance is generally difficult. In this context, it may be better to choose a classification algorithm like Bayesian networks and accept a lower TPR in cases where the mobile device does not afford sophisticated hardware. Indeed, Bayesian networks provides good detection rate and has a small memory footprint while being very fast at the same time. However, in cases where one affords a powerful mobile device, KNN or random forest should be his first choice. On the other hand, when our aim is to detect intruders taking as input user data coming from only one service, random forest is perhaps the best choice.

6. CONCLUSIONS AND FUTURE WORK

Modern mobile devices are capable of providing a wide range of services over several (mainly wireless) network access technologies. As a result of the frequent interaction between such devices and the Internet, a need for anomaly-based IDS is necessary. However, although a significant amount of work has been devoted to mobile device IDS in general, anomaly intrusion detection for such devices is still immature, and several problems remain unsolved. Our contribution is twofold. First, we try to evaluate and estimate the performance of four popular machine learning algorithms to detect misuse of mobile device based on user behavioural profiles. This is done in terms of TPR, accuracy and response time taking as input a dataset comprised from a satisfactory number of iPhone user data logs. Second, we examine the telephone call, SMS and Web browsing service logs not only separately but also combined in a multimodal fashion. This leads us to the creation of an integrated user behaviour profile that combines the most popular services so far. The ultimate goal here is to construct mobile user behavioural profiles for normal usage, with the purpose of alarming on user actions that deviate from the usual behaviour pattern. The results of the experimental procedure showed the ability of at least one algorithm to detect misuses with a very high success rate.

Currently, data analysis is done per service by taking into account important features of each data log. Another direction for future research is to organise the data into clusters, for example, per weekday or/and per week or even per hour, and perform additional experiments to further estimate the efficiency of such an IDS. Also, at present, we consider a proxy-assisted IDS system. That is, the application logic is divided between the mobile client and the proxy, which executes in the wired network and supports the client. This may be done to calibrate the algorithms and address the limitations of the portable device. From the knowledge gained, we are currently working towards extending this work by implementing a host-based anomaly IDS for ultramodern mobile devices. This will allow us to further study the effectiveness of such machine learning classifiers in terms of resource utilisation and speed of detection in real-time and directly on mobile hardware and software platform.

REFERENCES

1. Artail AH, Raydan M. Device-aware desktop Web page transformation for rendering on handhelds. *Personal and Ubiquitous Computing* 2005; **9**(6): 368–380. DOI: 10.1007/s00779-005-0348-5.
2. Chow GW, Jones A. A framework for anomaly detection in OKL4-Linux based smartphones. In *Proceedings of the Sixth Australian Information Security Management Conference*, Edith Cowan University, Perth, Western Australia, 1–3 December 2008.
3. Sun B, Chen Z, Wang R, Yu F, Leung VCM. Towards adaptive anomaly detection in cellular mobile networks. In *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC '06)*, Las Vegas, NV, USA, 8–10 January 2006, Vol **2**; 666–670.
4. Sun B, Xiao Y, Wu K. Intrusion detection in cellular mobile networks. In Wireless Mobile Network Security. Springer: Berlin, Heidelberg, 2007; 183–210. ISBN: 0387280405.
5. Naumann I, Hogben G, Fritsch L, Benito R, Dean R. Security issues in the context of authentication using mobile devices (Mobile eID). *ENISA Position Paper*. European Network and information Security Agency (ENISA), January. 2008.
6. GSMWorld Mobile. Market Data Summary (Q2 2009), 2009. Available at: http://www.gsmworld.com/newsroom/market-data/market_data_summary.htm [Accessed 16 February 2011].
7. Mobile World Congress. Visit Kaspersky Lab at Mobile World Congress 2009 in Barcelona, 2009. Available at: http://www.kaspersky.com/news?id=207575745 [Accessed 16 February 2011].
8. Landesman M. The World's Largest Security Analysis of Real-world Web Traffic. *Annual Global Threat Report*, ScanSafe STAT, 2009. Available at: http://www.scansafe.com/downloads/gtr/2009_AGTR.pdf [Accessed 16 February 2011].
9. Ray B. Home Office discusses thief-proof phones, 2007. Available at: http://www.theregister.co.uk/2007/05/25/home_office _phone_crime [accessed 16 February 2011].
10. Kruegel C, Valeur F, Vigna G. Computer security and intrusion detection. In Intrusion Detection and Correlation: Challenges and Solutions, Springer: Berlin, Heidelberg, 2005; 9–28.
11. Singh KK. Hybrid profiling strategy for intrusion detection, Department of Computer Science University of British Columbia, 2004.

12. Hammersland R. ROC in assessing IDS quality, Norwegian Information Security, Gjovik University College, 2007.
13. Moreau Y, Verrelst H, Vandewalle J. Detection of mobile phone fraud using supervised neural networks: a first prototype. In *Proceedings of the Seventh International Conference on Artificial Neural Networks (ICANN '97)*, Lausanne, Switzerland, 8–10 October 1997; 1065–1070.
14. Buschkes D, Kesdogan R, Reichl P. How to increase security in mobile networks by anomaly detection. In *Proceedings of the Computer Security Applications Conference*, Phoenix, 7–11 December 1998; 3–12.
15. Boukerche A, Notare MSMA. Behavior-based intrusion detection in mobile phone systems. *Journal of Parallel and Distributed Computing* 2002; **62**(9): 1476–1490.
16. Hollmén J. User profiling and classification for fraud detection in mobile communications networks. *PhD Thesis*, Helsinki University of Technology, 2000.
17. Burge P, Shawe-Tylor J. An unsupervised neural network approach profiling the behavior of mobile phone users for use in fraud detection. *Journal of Parallel and Distributed Computing* 2001; **61**(7): 915–925.
18. Sun B, Yu F, Wu K, Leung VCM. Mobility-based anomaly detection in cellular mobile networks. In *Proceedings of the ACM wireless security (WiSe'04)*, Philadelphia, PA, 1 October 2004; 61–69.
19. Kumpulainen P, Hätönen K. Anomaly detection algorithm test bench for mobile network management, Tampere University of Technology, 2008.
20. Bose A, Hu X, Shin KG, Park T. Behavioral detection of malware on mobile handsets. In *Proceedings of the Sixth International Conference on Mobile Systems, Applications, and Services (MobiSys'08)*, Breckenridge, CO, USA, 17–20 June 2008.
21. Liu L, Yan G, Zhang X, Chen S. VirusMeter: Preventing your cellphone from spies. In Proceedings of the 12th International Symposium on Recent Advances in Intrusion Detection, Lecture Notes In Computer Science, Springer-Verlag: Berlin, Heidelberg, 2009; 244–264. DOI: 10.1007/978-3-642-04342-0_13.
22. StatSoft. Neural networks, 2011. Available at: http://www.statsoft.com/textbook/stneunet.html [Accessed 16 February 2011].
23. Alpcan T, Bauckhage C, Schmidt AD. A probabilistic diffusion scheme for anomaly detection on smartphones. In *Proceedings of the 4th IFIP WG 11.2 International Workshop on Information Security Theory and Practices: Security and Privacy of Pervasive Systems and Smart Devices*, Passau, Germany, 12–14 April 2010; 31–46.
24. Damopoulos D, Kampourakis G, Gritzalis S. iSAM: An iPhone stealth airborne malware. In *Proceedings of the 26th IFIP TC-11 International Information Security Conference, IFIP AICT*, Springer: Berlin, Heidelberg, 2011; 17–28.
25. TiPB. 50 million iPhones sold +35 million iPod touches = 85 million iPhone OS devices, 2010. Available at: http://www.tipb.com/2010/04/08/50-million-iphones-sold-35-million-//ipod-touches-85-million-iphone-os//-devices [Accessed 16 February 2011].
26. Apple Inc. iPhone and iPod touch: about backups, 2011. Available at: http://support.apple.com/kb/HT1766 [Accessed 16 February 2011].
27. Peyravian M, Zunic N. Methods for protecting password transmission. *Computers & Security* 2000; **19**(5): 466–469.
28. Heckerman D. A Tutorial on Learning with Bayesian Networks. *Technical Report* 95–06, Microsoft Research Advanced Technology Division Microsoft Corporation, Redmond, USA, November 1995.
29. NeuroDimension. Radial basis function, 1998. Available at: http://www.nd.com/models/rbf.htm [Accessed 16 February 2011].
30. Wu X, Kumar V, Quinlan JR, et al. Top 10 algorithms in data mining. *Knowledge and Information Systems* 2008; **14**(1): 1–37.
31. Breiman L. Random forests. *Machine Learning* 2001; **45**(1): 5–32.
32. Schneider J. Cross validation, 1997. Available at: http://www.cs.cmu.edu/~schneide/tut5/node42.html [Accessed 16 February 2011].
33. The University of Waikato. Weka: Weka Machine Learning Project. Available at: http://www.cs.waikato.ac.nz/ml/weka [Accessed 16 February 2011].
34. Bouckaert RR. Bayesian network classifiers in Weka, 2004. Available at: http://weka.sourceforge.net/manuals/weka.bn.pdf [Accessed 16 February 2011].
35. Class RFNetwork. Weka class RBFNetwork, 2009. Available at: http://www.ia.udec.cl/~rzunigac/opinionApp/packages/weka-3-6-1/doc/weka/classifiers/functions/RBFNetwork.html [Accessed 16 February 2011].
36. Class IBk. Weka Class IBk, 2008. Available at: http://weka.sourceforge.net/doc/weka/classifiers/lazy/IBk.html [Accessed 16 February 2011].
37. Class RandomForest. Weka class random forest, 2007. Available at: http://weka.sourceforge.net/doc/weka/classifiers/trees/RandomForest.html [Accessed 16 February 2011].
38. Fawcet T. An introduction to ROC analysis. *Pattern Recognition Letters* 2006; **27**(8): 861–874.
39. Abouzakhar SN, Manson GA. Evaluation of intelligent intrusion detection models. *The International Journal of Digital Evidence* 2004; **3**(1): 1–20.