2014-06

Active authentication for mobile devices utilising behaviour profiling

Li, F

10.1007/s10207-013-0209-6
International Journal of Information Security
Springer Science and Business Media LLC

Active Authentication for Mobile Devices Utilising Behaviour Profiling

Fudong Li[1], Nathan Clarke[1,2], Maria Papadaki[1], Paul Dowland[1]

[1]*Centre for Security, Communications and Network Research (CSCAN), School of Computing & Mathematics, Plymouth University, Plymouth, PL4 8AA, United Kingdom*

info@cscan.org

[2]*School of Computer and Information Science, Edith Cowan University,*

Perth, Western Australia

Abstract
With nearly 6 billion subscribers around the world, mobile devices have become an indispensable component in modern society. The majority of these devices rely upon passwords and PINs as a form of user authentication and the weakness of these point-of-entry techniques are widely documented. Active authentication is designed to overcome this problem by utilising biometric techniques to continuously assess user identity. This paper describes a feasibility study into a behaviour profiling technique that utilises historical application usage to verify mobile users in a continuous manner. By utilising a combination of a rule-based classifier, a dynamic profiling technique and a smoothing function, the best experimental result for a user's overall application usage was an Equal Error Rate (EER) of 9.8%. Based upon this result, the paper proceeds to propose a novel behaviour profiling framework that enables user's identity verification through their application usage in a continuous and transparent manner. In order to balance the trade-off between security and usability, the framework is designed in a modular way that will not reject user access based upon a single application activity but a number of consecutive abnormal application usages. The proposed framework is then evaluated through simulation with results of 11.45% and 4.17% for the False Rejection Rate (FRR) and False Acceptance Rate (FAR) respectively. In comparison to point-of-entry based approaches, behaviour profiling provides a significant improvement in both the security afforded to the device and user convenience.

Keywords:
Active authentication, Behaviour profiling, biometrics

1 Introduction

With nearly 6 billion users globally, mobile devices have become ubiquitous in our daily life [17]. The modern mobile handheld device is capable of providing many multimedia services through a wide range of applications over multiple networks as well as on the handheld itself. These services are predominantly driven by data, which is increasingly associated with sensitive information. Indeed, a series of studies have highlighted the vulnerability of mobile devices by storing personal information (e.g. date of birth), login credentials (i.e. user name and password) and business data (e.g. corporate intellectual property) [20, 23, 30]. Therefore, any misuse of the services and information stored upon the device poses a threat to its owner when the device is lost or stolen, infected by malware [26], or attacked by social engineering [13].

The most popular user authentication approach is the password or Personal Identification Number (PIN). Although this approach is available on most mobile devices, a survey conducted by [9] demonstrated that 40% of their participants failed to utilise this simple security mechanism. In addition, possible misuse could still occur if mobile users do not utilise the technique properly, such as never changing the PIN, writing the PIN on a paper or sharing the PIN with others [6, 21]. However, the fundamental weakness of the PIN based approach is that as a point-of-entry technique it does not validate the user's identity again once the initial trust is obtained. DARPA proposed an Active Authentication program aiming to overcome the problem of the point-of-entry technique by utilising behavioural based biometrics for desktop computing [10]. According to the principle of the DARPA's proposal there are a number of biometrics that have the potential to be applied within the mobile device domain, such as gait recognition and keystroke dynamics. One such biometric approach is behavioural profiling. Its particular advantage being it is independent of the service being utilised (i.e. you need to be walking with gait and typing with keystroke analysis). Based upon the findings from a series of feasibility studies, this paper focuses upon demonstrating a novel behavioural profiling framework that can provide continuous and transparent protection for mobile devices.

This paper begins by introducing the study and presenting the state of the art in behavioural biometrics that have been applied within the mobile domain. The paper then proceeds to describe a comprehensive experimental study of mobile user application usage. Based upon the results, a novel behaviour profiling framework that will provide the verification of a mobile user's identity in a continuous and transparent manner is proposed and then evaluated through simulation. The paper concludes by highlighting the future direction of research.

2 Behavioural biometrics for mobile devices

The use of biometrics to authenticate a person\textquoteright s identity on mobile devices has been an established area for more than 10 years. In 1998, Siemens and Triodata developed a fingerprint recognition prototype with which a user can gain instant access to the phone by swiping their finger against a sensor. In 2011, facial recognition was utilised to unlock mobile users devices on the Samsung Galaxy Nexus smartphone [25]. However, these physiological biometric techniques are mainly deployed to offer point-of-entry security. In comparison, behavioural biometric methods have the ability to continuously and transparently verify user\textquoteright s identity. However, they also tend to have features that are noisier, vary more over time and are subject to various environmental aspects that can affect sample collection and classification. So care must be taken when considering their implementation particularly in active authentication where less control over the user and environment exists.

With the evolution of mobile devices, the chance to adopt other biometric based authentication approaches on them becoming more realistic. Indeed, some devices already possess a number of inbuilt sensors that are capable of collecting a variety of user biometric traits, enabling several behavioural approaches to be deployed upon them, including: behaviour profiling, gait recognition, handwriting recognition, keystroke analysis and voice verification. Plenty of research has been carried out for most of the aforementioned behavioural techniques [5, 7, 11, 31]. However, little work has been undertaken in the area of behaviour profiling.

Behaviour profiling identifies people based upon the way in which they interact with services of their mobile devices. In a behaviour profiling system, a user\textquoteright s current activities (e.g. dialling a telephone number) are compared with an existing profile (which is obtained from historical usage) by using a classification method (e.g. a Neural Network). The user\textquoteright s identity is determined based upon the comparison result. Therefore, behaviour profiling systems have the potential to provide continuous and transparent identification while users interact with their device.

Research into behaviour profiling started around 1997 by monitoring user calling and migration behaviour over service providers\textquoteright networks to detect telephony service fraud. An overview of studies and their performance characteristics are presented in Table 1. For call based fraud detection systems, a user\textquoteright s profile can be created based upon various calling features (e.g.
International Mobile Subscriber Identity (IMSI) and start date of call) that have been gathered over a service provider\textquoteright s network during a period of time [2, 15, 24]. The call based fraud detection system constantly monitors user\textquoteright s current calling activities again their profiles. An alarm will be raised if the deviation between the current calling activity and the user\textquoteright s profile exceeds a predefined threshold.

Based upon the theory that people have a predictable pattern when they travel from one location to another, the mobility based fraud detection system monitors a mobile user\textquoteright s location activities to detect abnormal behaviour (e.g. theft of the device) [3, 16, 28, 29]. By using user\textquoteright s historical location information that can be obtained either from a mobile cellular network in the form of cell IDs or via a Global Positioning System (GPS) link through latitudes and longitudes, user\textquoteright s mobility profile can be generated. Similar to the working principle of the call based fraud detection system an alarm will be raised if the user\textquoteright s current mobility activity significantly deviates from his/her profile.

Table 1 A review of network based mobile behaviour profiling

	Pattern classification model	DR	FAR
Calls	Mathematical formula [24]	80%	3%
Calls	Neural networks [15]	50%	0.02%
Calls	RBF neural network model [2]	97.5%	4.2%
Mobility	Bayes decision rule [3]	87.5%	-

Mobility	High order Markov model [29]	87.5%	15%
Mobility	Instance based learning [16]	50%	50%
Mobility	High order Markov model [28]	89%	13%

By monitoring a user\textquoteright s calling or location activities, behaviour based fraud detection systems offer high level of performance in terms of Detection Rate (DR) and False Alarm Rate (FAR) (as demonstrated in Table 1) and detect unforeseen attacks. Also, no additional computational power is required from the mobile device as the detection procedures are carried out over the service provider\textquoteright s network. Such solutions have been ideal for protecting traditional mobile devices in the terms of fraud as these devices cannot carry out the detection task by themselves and they are mainly utilised to access telephony and text message services. Nonetheless, as modern mobile devices have the ability to access multiple networks and host many applications, the behavioural based fraud detection systems cannot provide adequate protection for them anymore. Therefore, a new system that can offer protection for all networks and applications which mobile devices connect with and provide is needed. The next section will describe a feasibility study into behaviour profiling within mobile device host environment via user\textquoteright s application usage.

3 Behavioural Profiling on Mobile Devices

It is widely recognised that users utilise mobile services to perform a variety of tasks when interacting with individual applications. In order to thoroughly investigate the possibility of employing the behaviour profiling technique within the mobile host environment, two types of application behaviour will be examined for this research:

- Standard applications: provide a basic level of information on how a mobile user utilises the device, such as the name of an application, the time when the application was accessed and the location at which it was utilised.
- Extended applications: offer richer and more discriminatory information than standard applications do, i.e. information regarding what a user does with them (e.g. the telephone number within a telephone application and a web address within an Internet browser application).

The experiment employed a publicly available dataset that is provided by the Massachusetts Institute of Technology (MIT) Reality Mining project [12]. The MIT Reality Mining dataset contains a rich amount of mobile user\textquoteright s application activities over a long period of time: 106 participants enrolled for the data collection process from September 2004 to June 2005. The experiment utilised a subset of participants whose activities occurred during the period of 24/10/2004-20/11/2004 to maximise the number of participants. If two users had different start and end dates, the date feature alone would provide the discriminatory information required and skew the experimental result. These activities include 101 standard applications and two extended applications (telephony and Short Message Service (SMS)) as shown in Table 2. Despite some extended applications, such as an Internet browser application, were available in the dataset, the extra information (e.g. visited website) was not captured; hence, they were treated as standard applications. It was the only open dataset that contained sufficient quantity and quality of data to allow for the experiment to be conducted. By utilising the MIT dataset, three sets of experimental studies were conducted: descriptive statistics on mobile applications, a preliminary study on the telephony service and behaviour profiling on mobile applications. The first study sought to better understand the nature of the data with a view to identify possible features for subsequent analysis and evaluation. The second study was conducted to identify an optimal classifier for solving the behaviour profiling issue within the mobile host environment and to determine positive behaviour profiling features. The third study was designed to examine the feasibility of employing behaviour profiling on mobile devices through application activities.

Table 2: The MIT dataset

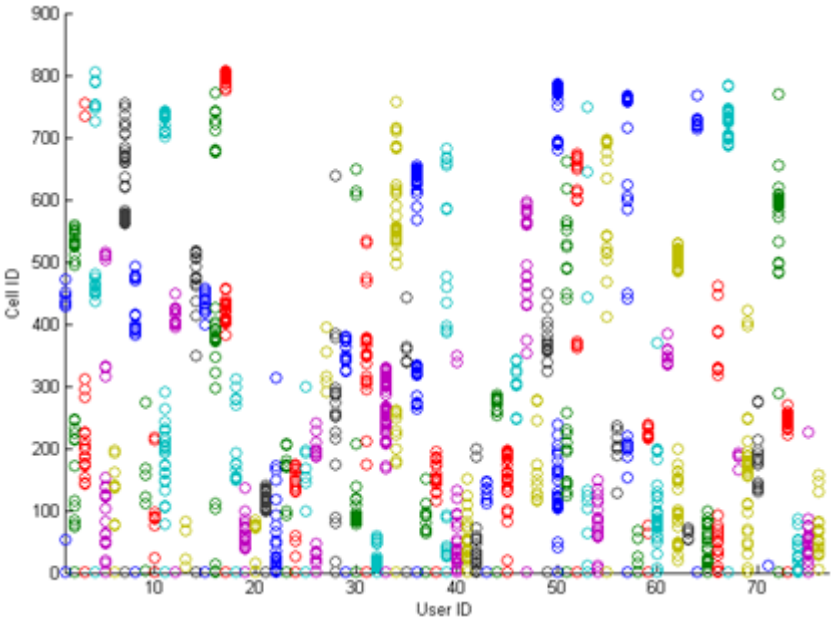
	Standard apps	**Telephony**	**SMS**
participants	76	71	22
unique apps/telephone numbers	101	2,317	1,381

logs	30,428	13,599	1,381

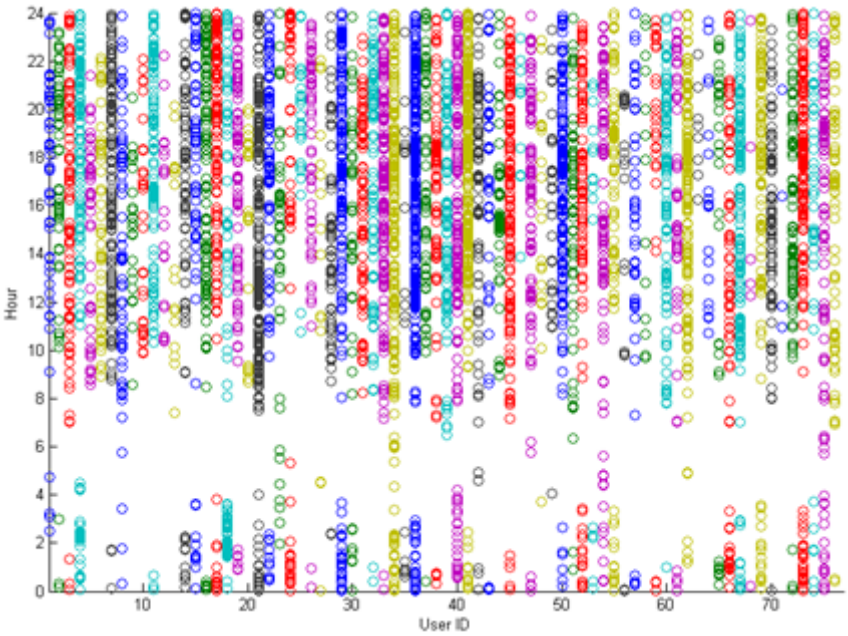
3.1 Descriptive Statistics Analysis

It has been established that descriptive statistics has the ability to determine potential positive features for forming unique patterns to discriminate individual users [18]. In order to examine potential positive features of both standard and extended applications for behaviour profiling, two sets of analysis were conducted on the data presented in Table 2.

For standard applications, their name, time and location of access features were available from the dataset and examined. For example, usages of all users\textquoteright phonebook application features are shown in Figure 1. For the location usage comparison, user 71 did not share any cells with any other users (although it is difficult to visualise due to the large cell IDs range in a relative small graph). As a result, this user could be identified based upon the location feature alone. On average, each user only shared 2 cells with another user. For the worst case, users 41 and 66 shared 10 cells. Although the usage on these 10 cells represent 70.8% and 62.5% of their total usage, they utilised these cells very differently: user 41 spent the majority of location usage in cells 1, 77 and 18; while user 66 utilised cell 49 most of the time. Therefore, the majority of their usage could still be separated. For the comparison on the time of accessing feature of the phonebook application, no clear patterns are shown between individual users in general despite the figure does highlight usage differences between several groups of users (i.e. high, medium and low).



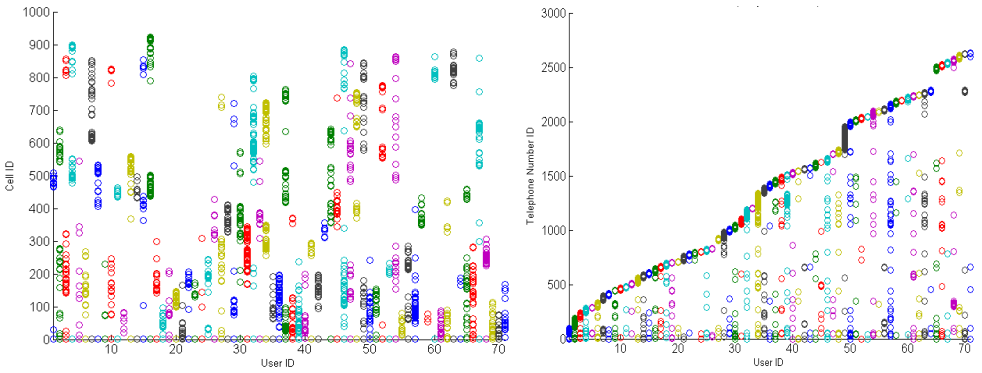
Location



Time

Figure 1: Usage comparison of all users\textquoteright phonebook features

For extended applications, the name of the application, time and location of access, plus extra information the application offers were available from the dataset and analysed. For example, Figure 2 depicts the usage comparison of the location and telephone number features of the telephony service from all 71 users. For usages comparison of the location feature similar patterns were presented as its counterpart of the phonebook application: individual users could be identified by utilising the location feature alone. For the comparison on the telephone number usage, among these users, user 26, 33 and 59 did not share any telephone number with any other users; also each user only shared one telephone number with another user on average. Hence, the majority of the users could be discriminated from each other based upon the telephone number feature alone. Nonetheless, some users did share several telephone numbers during the chosen period. For instance, user 34 and 63 shared 9 telephone numbers. However they could still be discriminated between each other as user 34 heavily used the anonymised telephone number 1237 while user 63 employed the anonymised telephone number 1247 and 1271 most of time. Regarding the time of call feature it is difficult to identify individual users by using it in isolation, similar to the situation for the phonebook application For the duration of call comparison, no clear discriminative information is available as the majority of calls lasted less than 300 seconds (i.e. 91.5% of 13,719 calls).



Location Telephone number

Figure 2: Usage comparison on 71 users\textquoteright telephony features

The statistical study demonstrates that the chosen dataset contains a huge amount of user\textquoteright s mobile applications activities. Although it could be seen that the majority of the users utilised their devices differently (especially for extended applications), a level of similarity was also observed for some users (i.e. the applications which they used and also where and when they utilised them). Moreover, a number of features have the potential to be used for discriminating mobile users. For the standard application, users could be separated via the name of individual applications if they did not utilise the same applications. For those who utilised the same applications, the majority of them could still be discriminated by knowing where the application was used. For the extended applications, apart from the location feature, the telephone number could also be very useful in distinguishing mobile users. However, by using the application activation time feature alone, it could be challenging to discriminate users from each other. This may be caused due to there only being 24 hours in a day: the longer the chosen period (i.e. more days), the higher the chance that two users will activate the same application within a similar time frame or even at the same time. In general, no clear difference is shown by the duration feature of telephony. There are many factors which may affect how long a call may last (e.g. the relationship between calling parties or how often they communicate), with the observed percentages it is clear that it could be difficult to discriminate users from each other by using the duration of call alone. A summary of the effectiveness of the aforementioned features towards a potential successful classification of mobile users is summarised in table 3.

Features	Contributions
Telephone number	Positive
Location	Positive
Duration of call	Negative
Time of call	Negative

Table 3: Individual application features towards to a successful classification

3.2 A preliminary study utilising telephony activity

The selection of effective features is a critical process in pattern classification as the system performance is closely related to the features and a large number of features will increase the complexity and the size of the classifier resulting in a problem known as the curse of dimensionality [1]. Also, the literature on behaviour profiling identified a number of classification methods that performed well and they fall into two categories: statistical and neural network approaches. Based upon the no free lunch theorems, there is no single classification method that can solve all given problems [31]. Three classification methods were chosen from these categories to identify an optimal classifier and explore the effectiveness of various application features to solve the behaviour profiling within the mobile host environment. These were the Radial Basis Function (RBF) neural network, the Feed-Forward Multi-Layered Perceptron (FF MLP) neural network and a Rule-based approach.

As the full dataset contains a total of 45,408 logs of applications activities, it could require a large amount of time to identify the usefulness of each application feature by employing the whole dataset. As the purpose of this experiment was merely to identify relevant features and classifiers rather than determining the performance, a sub-dataset that was extracted from the telephony activities containing a total of 3,836 logs of calls from 20 randomly selected users was utilised. For the actual experiment, each of these 20 users\textquoteright data was divided into two halves based upon the date of initiation feature: the first half was used to generate a profile and the other half was utilised to evaluate the performance of the classifiers and features in terms of False Acceptance Rate (FAR, False Rejection Rate (FRR) and Equal Error Rate (EER) (e.g. the lower the EER, the better performance). The results of the preliminary study are presented in the following subsections.

3.2.1 Radial Basis Function Network

An RBF neural network has been one of most popular pattern classification methods used in the Artificial Intelligence (AI) field [19]. By default, it has three network configurations: number of neurons, the performance goal and spread. For this preliminary study, only the number of neurons setting was configured while the other two remained as default. Table 4 demonstrates the results achieved by the optimal RBF neural network configurations with several combinations of telephony features as the inputs. By employing the dialled telephone

number and the location of call as the inputs with 75 neurons, the RBF network achieved the best performance with an EER of 10.5%.

Features Employed	Number of neurons			EER
1, 2, 3, 4	75			14.9%
1, 2, 4	100			14.5%
1, 2, 3	50			13%
1, 2	75			10.5%

Key: (1) Telephone No; (2) Location; (3) Duration; (4) Time

Table 4: The best RBF network configurations with various telephony features

3.2.2 Feed-Forward Multi-Layered Perceptron (FF MLP) Network

The FF MLP network is another widely employed AI technique utilised in the pattern classification domain [19]. With more network configuration variables available, the FF MLP neural network is a more complex classifier compared with the RBF neural network. For this preliminary study, apart from modifying the number of neurons parameter, other configurations of the FF MLP neural network were not altered. Table 5 demonstrates the best experimental results obtained by from several FF MLP network configurations with various combinations of telephony features. By using the dialled telephone number and the location of calling as the inputs and applying 150 neurons, the FF LMP classifier obtained its best performance producing an EER of 17.5%.

Features Employed	Number of neurons			EER
1, 2, 3, 4	125			21.3%
1, 2, 4	100			20.6%
1, 2, 3	150			24.7%
1, 2	150			17.5%

Key: (1) Telephone No; (2) Location; (3) Duration; (4) Time

Table 5: The best FF MLP network configurations with various telephony features

3.2.3 A rule-based approach

The basis for this approach was derived from the descriptive statistics produced when analysing the data and the large variances observed. A dynamic approach therefore seemed sensible to cope with the changing nature of the profile. Based upon the premise that the historical profile can be utilised to predict the probability of a current event, the rule-based approach illustrated in Equation 1 was devised. The approach provides a mechanism to ensure all outputs are bounded between 0 and 1 to assist in defining an appropriate threshold.

$$\text{Equation 1: Alarm if: } 1 - \frac{\sum_{i=1}^{N}\left(\frac{\text{Occurance of Feature}_{ix}}{\sum_{x=1}^{M}\text{Occurance of Feature}_{ix}}\right)}{N} \geq \text{threshold}$$

Where:

i=The features of one chosen application (e.g. dialled number for telephony application)
x=The value of Feature$_i$ (e.g. office telephone number and home telephone number)
M=Total number of values for Feature$_i$
N=Total number of features
Threshold= A predefined value according to each individual user

For the experiment, all parameters within the formula were chosen appropriately according to the selected telephony features. The results from the experiment are presented in Table 6. By using the dialled telephone number and location of calling features, the rule-based approach obtained a best result EER of 11%.

Features Employed			EER
1, 2, 3, 4			20.1%
1, 2, 4			12.4%
1, 2, 3			19.7%
1, 2			11%

Key: (1) Telephone No; (2) Location; (3) Duration; (4) Time

Table 6: Experimental results by employing the rule-based approach

The above three sets of results have demonstrated that the users can be discriminated by their telephony service within the mobile host environment with a good level of performance and also the usefulness of each application feature towards the classification result. Both the dialled telephone number and location of call features proved to be positive for discriminating the users by all three classifiers. In comparison, neither the time nor the duration of call feature can be considered as contributing positive information towards the classification process as by adding either or both of them as the inputs the performance of the classifiers decreased. In addition, the effectiveness of these features (i.e. location, telephone number, time and duration of call) was also suggested by the previous statistical analysis presented in section 3.1.

Among the three chosen classifiers, the FF MLP neural network achieved the worst performance while both the RBF neural network and the rule-based approach achieved much better performances. Despite the RBF neural network had a slightly higher performance (0.5% EER) it also consumed at least twice amount of computational power than the rule-based approach did (based upon observations during the experimental study). As a result, the rule-based approach was employed as the classifier with which to progress the next step of the research as its low requirements on computing power which is a key limited resource within the mobile environment.

3.3 Behaviour profiling on mobile applications

Based upon the findings from the descriptive statistics and preliminary studies, a complete experiment was undertaken on mobile user\textquoteright s application usage using the rule-based approach. In addition to the static profile technique, the experiment also utilised a dynamic profile technique to minimise the impact that is caused by users\textquoteright irregular mobile usage which was observed from the preliminary study (i.e. large FRR rates). The dynamic profile contains 7/10/14 days of each users most recent activities that was updated on a daily basis. The evaluation process for both static and dynamic profile techniques was carried out on the same dataset. A smoothing function that treats a number of successive applications activities as one event was also introduced to cope with the mobile user\textquoteright s inconsistent and variable usage behaviour; therefore, a decision is made based upon the combined events rather than a single occurrence. The following sections contain a complete behaviour profiling experimental study on mobile user\textquoteright s application usage considering standard, extended and multi-instance applications.

3.3.1 Standard applications profiling

For standard applications, the experiment employed all the standard applications activities that are presented in Table 2. The application name, date of initiation and location of usage features were extracted for generating profiles and validating the performance of the classifier. A complete set of experimental results for users\textquoteright standard application usage is presented in Table 7. The best performance (EER 13.5%) was obtained by utilising a 14 day dynamic profile with the smoothing function of 6 application entries (also shown in Figure 3).

	Number of application entries					
	1	2	3	4	5	6

	Static 14 days	21.1%	17.4%	16.3%	14.9%	14.2%	13.6%
	Dynamic 14 days	21.1%	17.3%	16.0%	14.5%	14.0%	13.5%
	Dynamic 10 days	22.1%	17.8%	16.2%	14.6%	14.4%	13.7%
	Dynamic 7 days	24.0%	19.4%	17.6%	15.9%	15.3%	14.4%

Table 7: Experimental results for standard applications

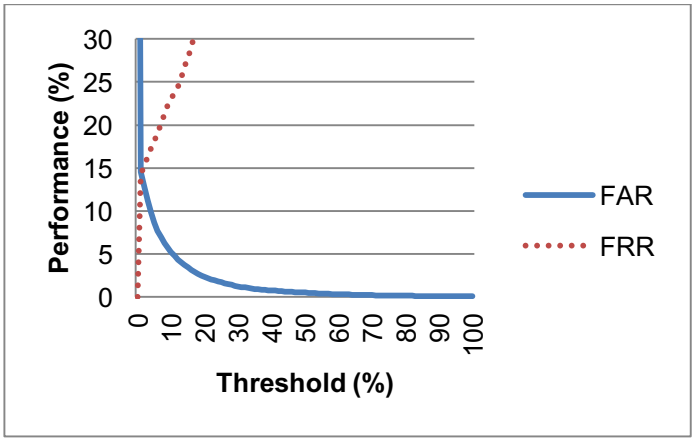


Figure 3: FAR-FRR plot for standard applications with the dynamic 14 day profile with 6 application entries

3.3.2 Extended applications

The telephone call experiment employed all the telephony activities as described in Table 2. The telephone number, date and location of call were extracted for the purpose of profile generation and data evaluation. A complete set of experiment results for users\textquoteright telephone call usage is shown in Table 8. The best experimental result for the users\textquoteright telephony activity is an EER of 5.4% and it was achieved by using the 14 day dynamic profile technique with the smoothing function of 6 telephone call entries (also depicted in Figure 4).

		Number of telephone call entries					
		1	2	3	4	5	6
	Static 14 days	9.6%	9.1%	7.9%	7.2%	4.3%	6.4%
	Dynamic 14 days	8.8%	8.1%	6.4%	6.4%	6.3%	5.4%
	Dynamic 10 days	9.6%	8.6%	8.1%	7.2%	6.9%	6.0%
	Dynamic 7 days	10.4%	8.8%	8.5%	7.3%	7.0%	6.2%

Table 8: Experimental results for the telephone call application

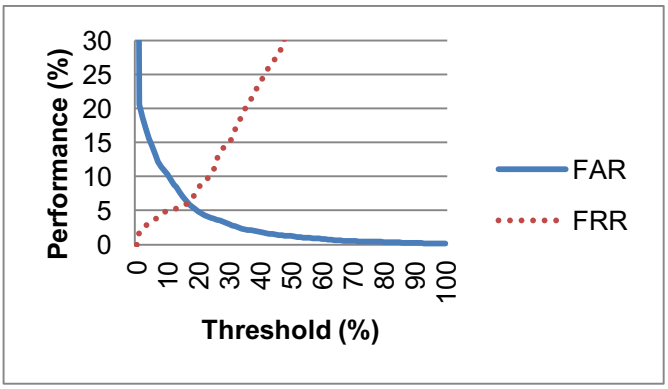


Figure 4: FAR-FRR plot for the telephone call application with the dynamic 14 day profile with 6 telephone call entries

The SMS experiment utilised the SMS message data as described in Table 2. For each SMS log entry, the texted telephone number, date and location of texting were extracted for building profiles and examining the performance of the classifier. As several participants had a limited number of messages over the chosen 28 day period, a maximum of 3 log entries were utilised for the smoothing function. The complete result for users SMS application usage is shown in Table 9. The best result is an EER of 2.2% and it was acquired by the classifier utilising the 14 day dynamic profile with a smoothing function of 3 text message entries (also shown in Figure 5).

		Number of text message entries		
		1	2	3
	Static 14 days	7.0%	4.3%	3.6%
	Dynamic 14 days	5.7%	2.6%	2.2%
	Dynamic 10 days	8.3%	4.1%	3.7%
	Dynamic 7 days	10.7%	5.7%	3.8%

Table 9: Experimental results for the SMS application

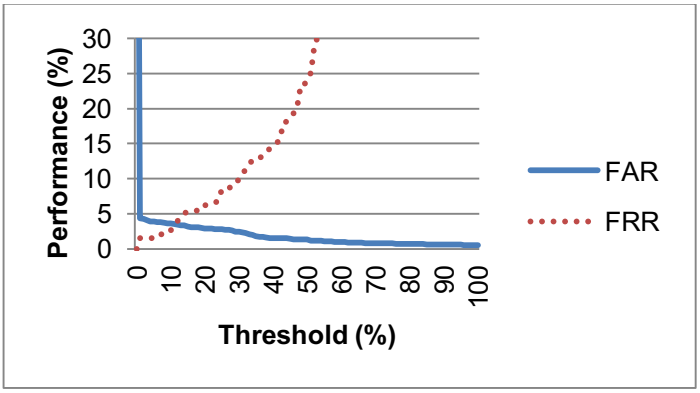


Figure 5: FAR-FRR plot for the SMS application with the dynamic 14 day profile with 3 text message entries

In daily life, mobile users utilise their applications in a chronological order. For instance, a user switches off the alarm clock (standard application) at 6:05 AM; at 7:10 AM, they make several phone calls (extended application) and start listening to music (standard application) at 7:36 AM. Therefore, the multi-instance applications

can continuously present an image of what a user does on the whole, while either the standard or extended applications could only partially provide information on user\textquoteright s activity. Hence, an experiment was conducted to examine the performance of the multi-instance applications technique for constantly monitoring every single activity to identify abnormal mobile usage.

For the multi-instance applications experiment, all users\textquoteright applications activities that are presented in Table 2 were utilised. For each user, their standard and extended applications were joined together by using the time stamp in a chronological order. Also, features were selected according to their application categories. In total, 30,428 standard and 15,101 extended applications logs were employed. The experimental results for user\textquoteright s multi-instance applications activities are demonstrated in Table 10. By utilising the dynamic profiling technique with 10 days of profiling data and a smoothing function with 6 log entries, the best result of EER 10% was obtained (also depicted in Figure 6).

		Number of log entries					
		1	2	3	4	5	6
	Static 14 days	16.9%	13.6%	12.7%	12%	10.9%	11%
	Dynamic 7 days	19 %	15.2%	13.1%	12.4%	11.3%	10.5%
	Dynamic 10 days	17.4%	13.7%	12.3%	11.6%	10.6%	9.8%
	Dynamic 14 days	16.5%	13.5%	12.1%	11.6%	10.5%	10.1%

Table 10: Experimental results for multi-instance applications

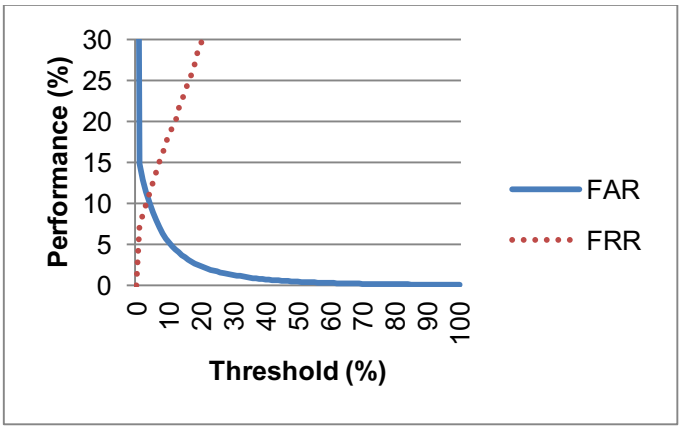


Figure 6: FAR-FRR plot for multi-instance applications with the dynamic 10 day profile with 6 application entries

3.3.3 Discussion

In general, the dynamic profiling technique achieved a slightly better performance than the static profiling technique did. This is reasonable as a dynamic profile contains a user\textquoteright s most recent activities rather than some usage that occurred several weeks ago and therefore is less relevant to user\textquoteright s current behaviour. Also, the performance is improved with a longer profile period. Hence, an increased number of days (e.g. 18/22 days) of user activities as a profile should be examined to find the optimum solution. Nonetheless, [14] suggests that users only keep 67% of new applications beyond a 30-day period indicating that users do change their usage pattern over time. While the smoothing function treated more application entries as one incident, the performance was also improved. The smoothing function reduces the impact any single event might have and seeks to take a more holistic approach to monitoring for misuse; this will provide a user-friendly environment as fewer rejections occur and more convenient when a user changes their usage behaviour. However, this approach does take a longer time for the system to make a decision providing a window of opportunity for misuse and a degree of abuse might be missed by the security control.

The name of application and location of usage are valuable features that can provide sufficient discriminatory information to individuate mobile users (as shown in Table 7). However, whilst this might identify many misuse scenarios, it would not necessarily identify all cases of misuse particularly those where a colleague might briefly misuse a device because the location information is likely to fall within the same profile as the authorised user. So care is required in interpreting these results. Limitations in the dataset are also likely to have created certain difficulties. Due to the small number of mobile applications that were available in 2004, a large similarity of application usage between users within the dataset creates difficulty for any classification method. However, it is envisaged that current mobile users application usage would arguably differ more as there are more than 1 million applications available in the mobile apps market. As a result, it would be arguably easier to discriminate mobile users through their application usage now.

The performance of telephony and the SMS applications are generally very good (as demonstrated in Tables 8 and 9) more than twice that of the standard application profiling. This reinforces the hypothesis that knowing both the application and what the user does with it, improves the chance of identifying individual users significantly. Moreover, mobile users had a far larger set of telephone contacts (the numbers they can dial/text) than applications, making the classification process easier with more identifiable data points from which to discriminate.

The experimental results of the multi-instance application (as presented in Table 10) are in between the results from the standard and extended applications; this is within the expectation as the experiment utilised the combination of both types. Based upon all experimental results, it is envisaged that the larger the proportion of extended applications users have, the better the performance of a system. As a result, the identification process of which category (either standard or extended) an application belongs to is mission critical for a behaviour profiling system.

Behavioural techniques	EER
Behaviour Profiling	10%
Gait recognition [11]	20.1%
Keystroke analysis [5]	13%
Handwriting recognition [7]	1%
Voice verification [32]	7.8%

Table 11: The performance comparison of behavioural techniques on mobile devices

The performance of the behaviour profiling technique is within the expectation of the overall behavioural biometric category in the mobile device environment (as illustrated in Table 11). However, in order to ensure this technique can be widely adopted by users, it would have not only to provide additional security but also user convenience. The next section of this paper describes a behaviour profiling framework that is designed to continuously profile and verify mobile user\textquoteright s identity in a transparent style.

4 A Novel Framework for Behaviour Profiling on Mobile Devices

The concept of transparently authenticating mobile device users was first proposed by the Transparent Authentication System (TAS), which utilises a mixture of biometric techniques to verify a mobile user\textquoteright s identity in a continuous and transparent manner [4]. Based upon the foundation laid by TAS, the behaviour profiling framework employs the behaviour profiling approach to provide an enhanced security for the mobile device with minimum user inconvenience and the framework works in the following manner:

- To improve the security for the mobile device beyond that offered by the password and token based approach;
- To verify the user based upon their application usage in a continuous style;
- To ensure the verification process is carried out in a user-friendly manner and that the user is mainly verified transparently rather than intrusively;

- To provide an architecture that can operate in one of three modes based upon the desired output implementation: as a standalone security control, within an IDS system as a misuse detector or within a TAS.

A number of process engines and a Security Manager have been devised to achieve these objectives (as illustrated in Figure 7). When a user utilises an application, details of the activity are automatically collected by a Data Collection Engine and then formulated into a behavioural sample. A Behaviour Classification Engine compares the sample with a profile that is pre-generated by a Behaviour Profile Engine to determine the legitimacy of the user. Based upon the verification result, the Security Manager can make one of the following decisions according to the mode in which the framework operates: for the standalone mode, the Security Manager individually handles the result and responses accordingly; otherwise, the Security Manager forwards the result to a security management system (e.g. TAS) that makes any final decisions. A detailed description of this process is presented in the following sections.

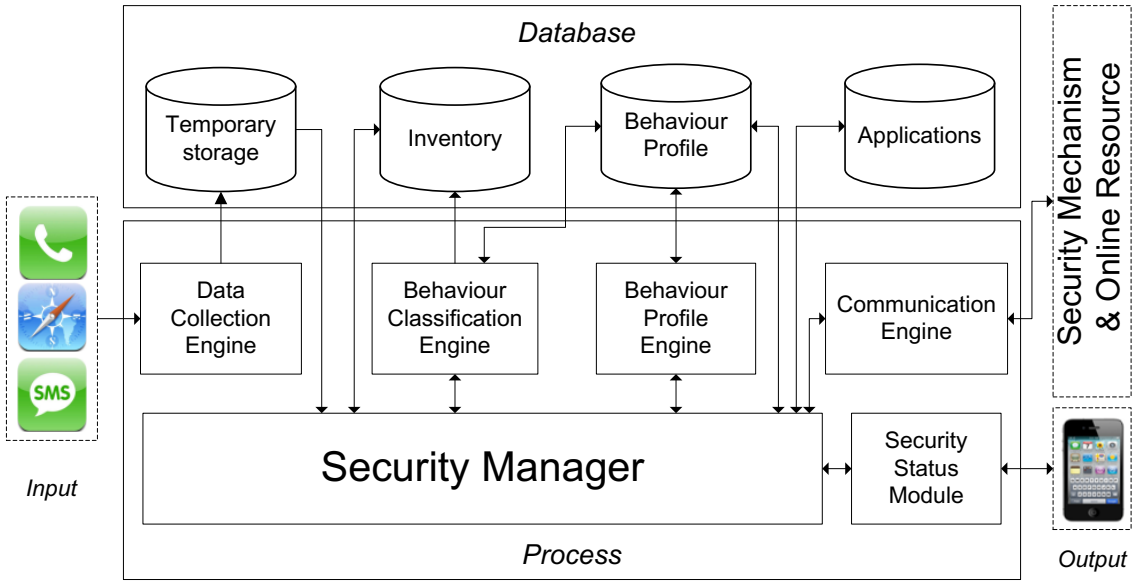


Figure 7: A novel Behaviour Profiling Framework

4.1 Processing Engines

The main duty of the Data Collection Engine is to capture a mobile user\textquoteright s application activities. When an application is utilised, the Data Collection Engine automatically gathers features associated with that application based upon the information presented in the applications storage. The features can be either related to the system level information of the application (e.g. the name of the application and location of usage) or the personalised user data (e.g. telephone numbers and email recipients).

The primary function of the Behaviour Profile Engine is to generate various templates by utilising the combination of the user\textquoteright s historical data, two template generation algorithms for standard and extended applications, a dynamic profiling technique and a smoothing function. The dynamic profiling technique is employed to maintain accuracy of the templates. While the smoothing function is utilised to provide a user-friendly environment.

The Behaviour Classification Engine provides the main functionality for the verification process when a verification requirement is met: the smoothing function, a verification time or the sensitivity of an application. The criteria of the smoothing function is that based upon the experimental results presented in section 3, a verification will not be performed unless a total of 6 applications have been utilised. The verification time is the

time period that controls how long the Behaviour Classification Engine has to wait to perform verification. Within the verification time, the requirement of the smoothing function is utilised as the reference for verification; otherwise, verification processes will be carried out regardless of the requirement of the smoothing function; this forces the framework to always provide security within the reference timeframe. The sensitivity of the application is a measure of the value of the application and/or its data. According to [22], a high sensitive mobile application is associated with high-risk levels. When a user requests access to a high sensitive mobile application but the current security status of the device is below the requirement for accessing it, the Behaviour Classification Engine will verify any applications which have not been verified and update the security status even though neither of the aforementioned requirements is met.

The Communication Engine acts as a gateway for the framework. By utilising it, the framework can download application features information from an online repository to facilitate the data collection process. When the framework operates in standalone mode and the device is locked down, the Communication Engine sends a code to the user by using which they unlock their device. While when the framework operates in dependent mode, the communication engine works as a bridge between the framework and a comprehensive security management system by forwarding verification results and accepting commands to/from the security management system.

4.2 Security Status Module

The main function of the Security Status Module is to maintain the System Security Status (SSS) level that constantly indicates how secure the system is. By utilising the SSS level, the framework can provide or deny access to the user accordingly. The SSS level is a numeric value in the range of -3 to +3[1]: -3 indicates low security whilst +3 demonstrates high security; it is calculated based upon two critical factors: the performance factor of an application and the verification result. The performance factor is dynamically allocated to each application based upon their performance as demonstrated in Table 12.

Table 12 Application Performance Factor

Application Performance (EER)	Factor
0-2%	1
2-4%	0.9
4-6%	0.8
6-8%	0.7
8-10%	0.6
10-12%	0.5
12-14%	0.4
14-16%	0.3
16-18%	0.2
>18%	0.1

After the activity of an application is verified, a temporary value will be allocated based upon its performance factor. When a verification process involves more than one application, the temporary value will be obtained by combining the performance factor of each individual application. Depending upon the verification result, the temporary value is then added to (verified successfully) or subtracted from (verified unsuccessfully) the existing SSS level to derive the current SSS level as shown in Figure 8.

$$Current\ SSS\ level = \sum_{i=1}^{m} VR \times APF_i + Existing\ SSS\ level$$

Where VR=Verification result, APF= Application Performance Factor, m= number of applications

[1] The boundaries defined on the numerical scale are only provided as a suggestion.

Fig. 1. The SSS level calculation process

\subsection{}

$$Degradation\ function = Current\ SSS\ level - \frac{Time\ elapsed}{Time\ Period}$$

Where Current SSS level > 0;
When (Current SSS level * Time period) <Time elapsed, SSS level equals 0;

Fig. 2. The SSS level degradation function

4.3 Security Manager

The Security Manager is the brain of the framework as it co-operates with other elements to complete various tasks, including continuously verifying the user\textquoteright s identity, updating the performance factor of an application and maintaining the SSS level.

The key task of the Security Manager is to monitor the current SSS level and make subsequent decisions accordingly when the user requests access to an application. This is achieved by utilising an SSS Monitor And Response (SMAR) algorithm which is the core security component of the proposed framework (as illustrated in Figure 10). The algorithm contains three main checking
stages before the device is locked down. These checking stages were chosen to provide a high level of user convenience and improved security. The algorithm employs a mixture of transparent and intrusive methods to verify user\textquoteright s identity. However, it is envisaged that the majority of legitimate users will experience transparent phases; while intrusive verification challenges are only utilised to ensure a user\textquoteright s legitimacy in the event of access being required to the mobile device but the SSS level being below security requirements.

When a user requests access to an application, the Security Manager initially examines the sensitivity of the application (either high sensitive or non-sensitive). A high-level security requirement is set for high sensitive applications as they are associated with critical information: they cannot be accessed when the SSS level is below 2 unless the user passes an intrusive verification. In comparison, the security requirement for non-sensitive applications is much more relaxed: they can be utilised when the SSS level is greater or equal to -2; otherwise, an intrusive verification will be used before the application can be accessed.

For the second checking stage, the Security Manger compares the current SSS level with the security requirements. If the requirements are met, the users will be granted access to the application. Otherwise, further checking will be performed based upon the nature of the application. If the application is a non-sensitive application, the user will be challenged with a randomly selected security question; if the application is associated with high sensitive information, the Security Manger will utilise the third verification requirement as described earlier and perform any subsequent actions.

For the final checking stage, the Security Manger utilises a randomly selected security question as the intrusive verification method to verify the user\textquoteright s identity. An intrusive flag that indicates how many times the intrusive verification method is utilised increases by 1 when the user answers the question; by default, the value of the intrusive flag is 0. The user will be granted access to the application if the question is correctly answered. Also, the current SSS level will be increased by a value (e.g.
1) and the intrusive flag will be set to 0. Otherwise, the intrusive flag remains the same and the user will be challenged again if the value of the flag is not greater than 2. When the value of the flag equals 3, the device will be locked down and only an administrative security password can unlock it. When a correct unlock code is entered, both the value of the intrusive flag and the SSS level will be set to 0 and the user will be able to access the device once again.

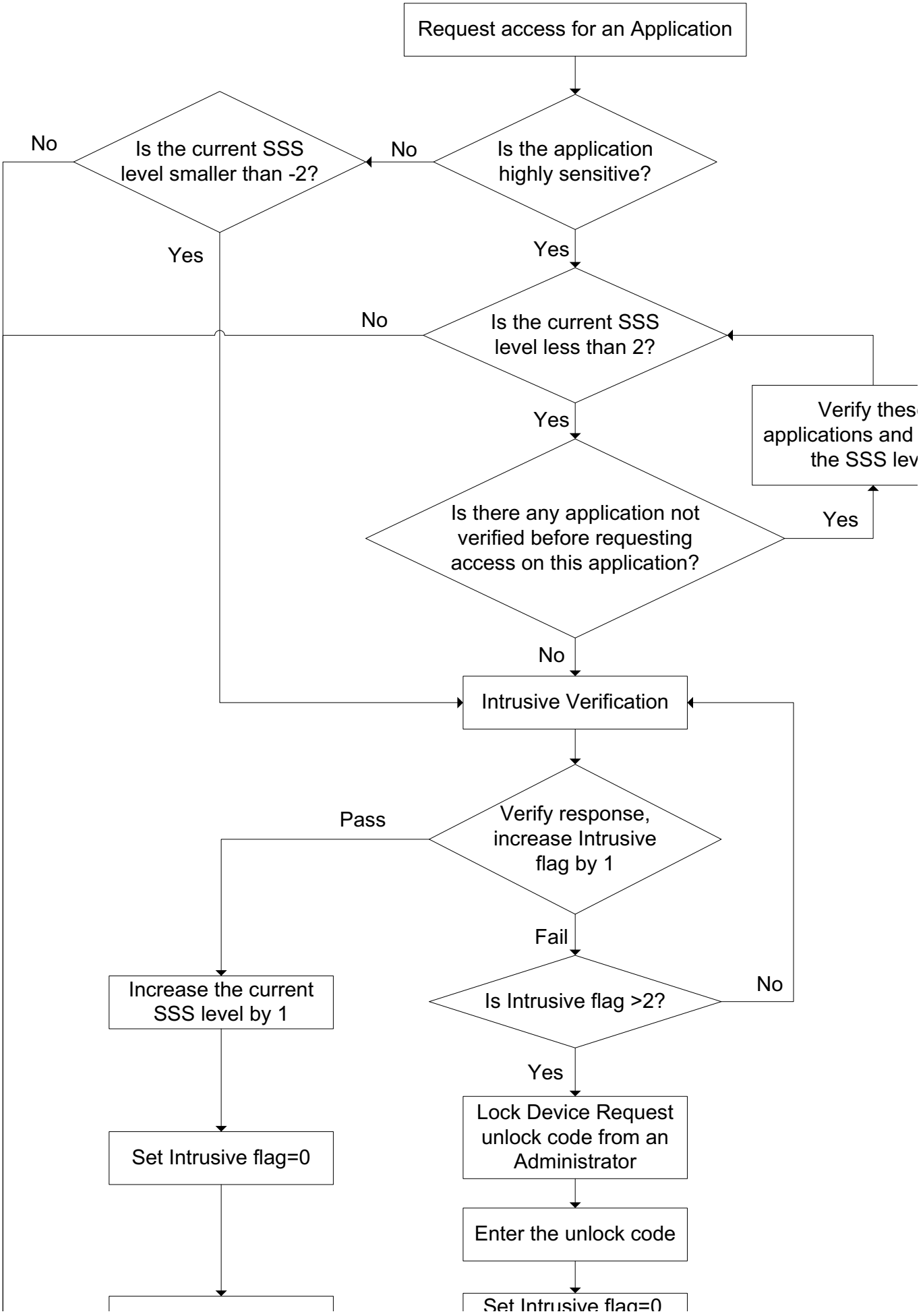

Request access for an Application
Is the application highly sensitive?
No
Is the current SSS level smaller than -2?
No
Yes
Yes
Is the current SSS level less than 2?
No
Yes
Verify thes applications and the SSS lev
Is there any application not verified before requesting access on this application?
Yes
No
Intrusive Verification
Verify response, increase Intrusive flag by 1
Pass
Fail
Is Intrusive flag >2?
No
Yes
Increase the current SSS level by 1
Set Intrusive flag=0
Lock Device Request unlock code from an Administrator
Enter the unlock code
Set Intrusive flag=0

Fig.10. Security Manager: SSS Monitor And Response algorithm

The three-stage SMAR algorithm of the Security Manager is the core component of the framework to adjust the balance between user convenience and system security. For non-sensitive applications, the user convenience is achieved by allowing users to have at least two transparent chances (depends upon the performance of individual applications) to verify themselves correctly before being intrusively authenticated. As a result, it is envisaged that if the system is working correctly, the SSS level should be high enough to permit automatic access for legitimate users and their experience of intrusive security challenges will be minimal. Nonetheless, the same configuration would also permit potential misuse on mobile applications as long as the SSS level is above the threshold (i.e. not smaller than -2). This phenomenon is difficult to avoid for any behavioural based biometric due to the trade-off between the FAR and the FRR. However, if the imposter continuously abuses mobile services, the probability of misuse that goes under the radar of the framework is getting smaller as eventually the SSS level will be reduced to the threshold and the device will be locked down. For sensitive applications, due to the higher security requirement for them, the chance of users being intrusively verified at the beginning of usage is significantly higher compared with when they access non-sensitive applications. This intrusiveness will be gradually reduced as legitimate users continuous activities will keep the SSS level high enough allowing automatic access to sensitive applications. However, the chance of imposters accessing sensitive applications is virtually impossible as their activities likely differ from legitimate users profile forbidding the SSS level to satisfy the requirements for access to sensitive applications.

When the framework works in the dependent mode, it can become a component for an authentication security mechanism (e.g. TAS) or an IDS security mechanism (e.g. the Knowledge-based Temporal Abstraction (KBTA) method [25]) to complete these mechanisms and enhance their ability and performance. Therefore, the Security Manager only provides a verification result and the final decision will be made by the other security mechanisms. As a result, it would be difficult to evaluate the performance and impact of the framework on other security mechanisms when it operates in the dependent mode.

5 Evaluation

In order to understand the effect that the framework has upon the overall performance, two aspects of the framework should be examined: the impact on the processing power and the effectiveness of authenticating the user. Regarding consumption of the processing power, previous research demonstrates that a complicated multimodal biometric authentication system (i.e. TAS) was prototyped within the mobile environment and users were satisfied with their performance [8]. Therefore, it is envisaged that the proposed framework will have a small processing power footprint and little effect on the performance of the mobile device. For the performance of authentication, the framework was evaluated via a simulation process which was conducted within the Matlab environment. The simulation system employed the same 76 users\textquoteright 4 weeks mobile activities which were utilised in section 3 as the simulation data. For each user, their activities were divided into two halves, containing first and second two-week activities respectively. A user\textquoteright s profile was initially trained by utilising the first two-week worth of activities; with the profile then being updated on a daily basis. The rest of users\textquoteright activities were employed to evaluate the performance of the Behaviour Profiling framework. Furthermore, due to the lack of sensitive application usage within this dataset, the text message application was selected as a sensi-

tive application in order to evaluate the effect upon the framework. During the chosen period, 22 users utilised the text message application, representing 4.3% of the total application usage.

As discussed in section 4, the performance of the framework can be influenced by three key parameters: the smoothing function, the verification time and the degradation function. Therefore, the evaluation sought to analyse the effect these parameters have upon the performance. As such, four scenarios were set up to independently assess each parameter in turn, as illustrated in Table 13. Time periods for the degradation function were set between 1-60 minutes with a 10-minute interval for all four scenarios.

Table 13. Four scenarios for the simulation process

Scenario A	Smoothing function: 1 application; Verification time: NA
Scenario B	Smoothing function: 3 applications; Verification time: 3 minutes
Scenario C	Smoothing function: 3 applications; Verification time: 6 minutes
Scenario D	Smoothing function: 6 applications; Verification time: 6 minutes

5.1 Simulation results

The framework was configured according the setup in Table 13. Based upon the verification requirement of the Behaviour Classification Engine (in section 4), a user\textquoteright s application activity will be verified as soon as one application is utilised; therefore, the verification time is not applicable. The simulation results for scenario A are presented in Table 14.

Table14 Simulation results of scenario A

	Non-sensitive apps		**Sensitive apps**		**Overall apps**	
	FRR	**FAR**	**FRR**	**FAR**	**FRR**	**FAR**
1 min	9.05	4.35	98.89	0	12.91	4.17
10 mins	8.97	4.35	91.76	0	12.53	4.17
20 mins	8.88	4.36	85.9	0	12.19	4.17
30 mins	8.82	4.36	83.04	0	12.01	4.17
40 mins	8.71	4.36	79.87	0	11.77	4.17
50 mins	8.66	4.36	76.86	0	11.59	4.17
60 mins	8.6	4.36	74.8	0	11.45	4.17

In comparison with the configuration of scenario A, scenario B employed more applications for the smoothing function, allowing the smoothing function to work with up to 3 application activities. According to the verification requirements of the Behaviour Classification Engine, within the 3 minutes verification time window, applications will be processed when the total number being utilised reaches 3. Otherwise, any utilised applications will be verified even the total number of them is smaller than 3. Table 15 demonstrates the simulation results for scenario B.

Table 15 Simulation results of scenario B

	Non-sensitive aps		**Sensitive apps**		**Overall apps**	
	FRR	FAR	FRR	FAR	FRR	FAR
1 min	8.05	3.41	100	15.29	13.06	4.07
10 mins	7.87	3.42	94.28	15.29	12.53	4.08

20 mins	7.75	3.42	88.8	15.29	12.08	4.08
30 mins	7.72	3.42	85.95	15.29	11.88	4.08
40 mins	7.64	3.42	81.89	15.29	11.58	4.08
50 mins	7.62	3.42	78.57	15.29	11.38	4.09
60 mins	7.57	3.42	77	15.29	11.24	4.09

Based upon the setup of scenario B, scenario C utilised a longer verification time; this increases the potential for allowing more application activities to be processed within one smoothing function. Based upon the requirement of the Behaviour Classification Engine, application activities will be classified as soon as the total number of them reaches 3 within the 6 minutes verification time window; when the 6 minutes verification time window is surpassed, even if the total number of application activities is smaller than 3, they will be processed. Simulation results of scenario C are presented in Table 16.

Table 16 Simulation results of scenario C

	Non-sensitive apps		**Protected apps**		**Overall apps**	
	FRR	**FAR**	**FRR**	**FAR**	**FRR**	**FAR**
1 min	7.96	2.45	100	26.39	13.3	3.95
10 mins	7.83	2.46	96.23	26.39	12.89	3.96
20 mins	7.66	2.46	90.64	26.39	12.36	3.96
30 mins	7.63	2.46	87.8	26.39	12.15	3.96
40 mins	7.54	2.47	84.07	26.39	11.79	3.96
50 mins	7.51	2.47	80.85	26.39	11.58	3.97
60 mins	7.45	2.47	79.26	26.39	11.43	3.97

In comparison with the setup for scenario C, scenario D employed a higher number of applications for the smoothing function; this allows the smoothing function to potentially work with up to 6 application activities. According to the verification requirements for the Behaviour Classification Engine, within the 6 minutes verification time, application activities will be classified once there are 6 applications being utilised. When the 6 minutes verification time is exceeded, application activities will be processed by the Behaviour Classification Engine even though the total number of activities is less than 6. The simulation results for scenario D are presented in Table 17.

Table 17 Simulation results of scenario D

	Non-sensitive apps		**Protected apps**		**Overall apps**	
	FRR	**FAR**	**FRR**	**FAR**	**FRR**	**FAR**
1 min	7.97	2.48	100	26.73	13.58	4.04
10 mins	7.85	2.49	96.19	26.73	13.2	4.04
20 mins	7.77	2.49	90.71	26.73	12.7	4.05
30 mins	7.69	2.49	87.46	26.73	12.42	4.05
40 mins	7.58	2.49	84.08	26.73	12.03	4.05
50 mins	7.57	2.5	80.76	26.73	11.82	4.05
60 mins	7.49	2.5	78.86	26.73	11.63	4.05

5.2 Discussion

In order to evaluate the performance of the framework, the most widely utilised PIN based technique was chosen as a baseline method. In order to maximise the security, it is assumed that a PIN is required after the mobile device has been idle for more than one minute. By utilising this setting, the PIN

based method was applied to the same simulation data, requiring users to enter a PIN for every single application usage (0% transparent authentication). In comparison, taking scenario A as it is the most similar configuration to the PIN-based approach shows the Behaviour Profiling framework achieved an overall FRR of 11.45%, indicating that 88.55% of the time the legitimate user will be transparently verified and automatically obtain access to the device. With the same configuration, the imposter has only got a 4.17% chance to misuse an application and conversely 95.83% of the time they will be denied access. It is worth noting the above simulation results did not include the intrusive stage of the authentication process. Therefore, in reality, it is highly likely that the imposter will be intrusively prompted with a randomly selected security question and hence the device will be locked down, resulting in an improved overall FAR. Arguably, the Behaviour Profiling framework is capable of providing continuous and transparent protection for a good proportion of the time and is able to do so in a more secure and user convenient fashion.

The smoothing function, verification time and degradation function are employed to justify the balance between the user\textquoteright s convenience and system security and their impact were also examined through the simulation scenarios. As demonstrated by the simulation results (from Table 14-17), the best system performance (11.24% for overall FRR and 4.09% for overall FAR) was obtained by utilising a combination of the smoothing function set to 3 applications, a verification time of 3 minutes and degradation function of 60 minutes (scenario B). With other configurations, the overall system performance decreased slightly. What the simulation results have shown is the effect these parameters have upon the framework and the need to ensure they are set on an individual rather than population basis so that the system can be configured to optimally perform given an individual users\textquoteright behaviour profile.

6 Conclusions and Future Work

The first part of this paper demonstrated a feasibility study which showed that by utilising a number of classifiers, mobile device users can be discriminated from each other based upon their application usage. The rule based approach proved the most suitable classifier for mobile device users in terms of performance and computational power. The dynamic profiling and smoothing techniques were also put in place to cope with the inconsistency of user behaviour and as a result better performance was obtained.

Based upon the promising experimental result, the second part of this paper proposed the Behaviour Profiling framework and described its core components and functionalities. The framework is able to provide a continuous and non-intrusive verification mechanism in standalone, TAS or IDS modes. By monitoring users application activities, the SMAR algorithm can continuously evaluate the system security status based upon which the framework can a decision whether to grant the access to users. The framework is subsequently analysed utilising a real user dataset in a set of scenarios, providing conclusive evidence that it can provide transparent identity verification a good proportion of the time - thereby outperforming traditional PIN-based authentication. However, the work is still required on improving the level of accuracy.

Future work will focus into two directions. Firstly upon developing a fully functioning prototype of the proposed framework so that a series of studies examining the practical usability of the approach can be measured. Secondly to undertake a wide-spread data collection activity so that a modern and relevant corpus of behavioural-based data exists from which algorithms and models can be tested and evaluated.

References

1. Bishop, M, Neural Networks for Pattern Classification,
Oxford University Press (1995)

2. Boukerche, A., Nitare, M.S.M.A, Behavior-Based Intrusion Detection in Mobile Phone Systems, Journal of Parallel and Distributed Computing, 62(9), 1476-1490 (2002)
3. Buschkes, R., Kesdogan, D., Reichl, P., How to increase security in mobile networks by anomaly detection, In Proceedings of the 14th Annual Computer Security Applications Conference, 3-12 (1998)
4. Clarke, N., Transparent User Authentication, Springer (2011)
5. Clarke, N.L., Furnell, S.M., Authenticating Mobile Phone Users Using Keystroke Analysis, International Journal of Information Security, 6(1), 1-14 (2006)
6. Clarke, N.L., Furnell, S.M., Authentication of users on mobile telephones - A survey of attitudes and practices, Computers & Security, 24(7), 519-527 (2005)
7. Clarke, N.L., Mekala, A.R., The application of signature recognition to transparent handwriting veri_cation for mobile devices, Information Management & Computer Security, 15(3), 214-225 (2007)

Clarke, N.L., Karatzouni, S., Furnell, S.M., **Flexible and Transparent User Authentication for Mobile Devices**, Proceedings of the 24th IFIP TC 11 International Information Security Conference, Pafos, Cyprus, May 18-20, ISBN: 978-3-642-01243-3, pp1-12, 2009
9. Credant, Phone Data makes 4.2 Million Brits Vulnerable to ID Theft, Credant, http://www.credant.com/news-a-events/press-releases/337-phone-data-makes-42-million-brits-vulnerable-to-id-theft.html (2009), Accessed: 14 June 2012
10. DARPA, Active Authentication, DARPA, http://www.darpa.mil/Our Work/I2O/Programs/Active Authentication.aspx (2011), Accessed: 17 April 2012
11. Derawi, M.O., Nickel, C., Bours, P., Busch, C., Unobtrusive User-Authentication on Mobile Phones Using Biometric Gait Recognition. In Proceedings of the 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp.306-311 (2010)
12. Eagle, N., Pentland, A., Lazer, D., Inferring Social Network Structure using Mobile Phone Data, In proceedings of the National Academy of Sciences (PNAS), 106, 15274-15278 (2009)
13. FBI, Smishing and Vishing, The FBI, http://www.fbi.gov/news/stories/2010/ november/cyber 112410/cyber 112410 (2010), Accessed: 11 April 2012
14. Flurry, Mobile Apps: Models, Money and Loyalty. Flurry Smartphone Industry Pulse, http://blog.urry.com/bid/26376/Mobile-Apps-Models-Money-and-Loyalty (2009), Accessed: 01 August 2012
15. Gosset, P, ASPeCT: Fraud Detection Concepts: Final Report. Doc Ref. AC095/VOD/W22/DS/P/18/1 (1998)
16. Hall, J., Barbeau, M., Kranakis, E., Anomaly-based intrusion detection using mobility pro_les of public transportation users, In Proceeding of IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, 2, 17-24 (2005)
17. ITU, Key Global Telecom Indicators for the World Telecommunication Service Sector, International Telecommunication Union, http://www.itu.int/ITU-D/ict/statistics/at glance/KeyTelecom.html (2011), Accessed 01 April 2012
18. Jain, A.K., Duin, R.P.W., Mao, J, Statistical pattern recognition: a review. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22(1), 4-37 (2000), doi: 10.1109/34.824819
19. Jain, A.K., Mao, J., Mohiuddin, K.M., Arti_cial neural networks: a tutorial, Computer, 29(3), 31-44 (1996), doi: 10.1109/2.485891
20. Kaspersky Lab, European Users Mobile Behaviour and Awareness of Mobile Threats, Kaspersky Lab ZAO, http://www.kaspersky.com/news?id=207576289 (2011), Accessed: 25 May 2012
21. Kurkovsky, S., Syta, E., Digital natives and mobile phones: A survey of practices and attitudes about privacy and security, In Proceedings of the IEEE International

Symposium on Technology and Society (ISTAS) , 441-449 (2010)
22Ledermuller, T., Clarke, N.L., Risk Assessment for Mobile Devices, In proceedings of Privacy and Security in Digital Business 8th International Conference, TrustBus , 210-221 (2011)
23. Power, R., Mobility and Security: Dazzling Opportunities, Profound Challenges, McAfee, http://www.mcafee.com/us/resources/reports/rp-cylab-mobile security. pdf (2011), Accessed: 1 May 2012
24. Samfat, D., Molva, R., IDAMN: an Intrusion Detection Architecture for Mobile Networks, IEEE Journal on Selected Areas in Communications, 15(7), 1373-1380 (1997)
25. Samsung., Galaxy Nexus, Samsung, http://www.samsung.com/uk/consumer/ mobile-devices/smartphones/android/GT-I9250TSAXEU (2012), Accessed: 04 May 2012
26. Securelist, Mobile Malware Evolution: An Overview, Part 3, Securelist, http://www.securelist.com/en/analysis?pubid=204792080 (2009), Accessed: 30 March 2012
27. Shabtai, A., Kanonov, U., Elovici, Y., Intrusion detection for mobile devices using the knowledge-based, temporal abstraction method. J. Syst. Softw., 83(8), 15241537 (2010)
28. Sun, B., Chen, Z., Wang, R., Yu, F., Leung, V.C.M., Towards adaptive anomaly detection in cellular mobile networks, In the IEEE Consumer Communications and Networking Conference, 2, 666-670 (2006)
29. Sun, B., Yu, F., Wu, K., Leung, V.C.M., Mobility-based anomaly detection in cellular mobile networks, Proceedings of ACM wireless security (WiSe 04), 61-69 (2004)
30. Which?, 13.5 million UK mobile phone users at risk of fraud, Which? Tech Daily, http://blogs.which.co.uk/mobile/mobile-phones/13-5-million-uk-mobile-phone-users-at-risk-of-fraud/ (2011), Accessed: 31 July 2012
31. Wolpert, D.H., Macready, W.G., No Free Lunch Theorems for Optimization, IEEE Transactions on Evolutionary Computation, 1, 67-82 (1997)
32. Woo, R., Park, A., Hazen. T., The MIT Mobile Device Speaker Veri_cation Corpus: Data collection and preliminary experiments, In Speaker and Language Recognition Workshop, 1-6 (2006)