Mobile User Signature Extraction based on user behavioural Pattern (MUSEP)

Hassan Sbeyti[1] Beatrice El Hage[1] and Ahmad Fadlallah[12]

[1]Arab Open University,Omar Bayhoum Street, Beirut, Lebanon
{hsbeity,bhage}@aou.edu.lb,
http://www.aou.edu.lb
[2]University of Sciences and Arts in Lebanon, Airport Road, Beirut, Lebanon
a.fadlallah@usal.edu.lb,
http://www.usal.edu.lb

Abstract. Purpose - The purpose of this paper is to extract the user behavior and transform it into a unique signature that can be used as implicit authentication techniques. Smart devices are equipped with multiple authentication techniques and still remain prone to attacks since all of these techniques require explicit intervention of the user. Entering a pin code, a password or even having a biometric print can be easily hacked by an adversary.
Design/methodology/approach - In this paper, we introduce a novel authentication model that is intended to be used as complementary to the existing models; Particularly, the duration of usage of each application and the occurrence time were examined and modelled into a user signature. During the learning phase, a cubic spline function is used to extract the user signature based on his behavioral pattern.
Findings - Preliminary field experiments show a 70% accuracy rate in determining the rightful owner of the device.
Originality/value - The main contribution of this work is a framework that extract the user behavior and transform it into a unique signature that can be used to implicitly authenticate the user.

Key words: Security, implicit authentication, behavioural modelling
Paper type Research paper

1 Introduction

The technological advances in all domains are making the use of smart devices in everyday life more imposing. These range from smart phones to laptops, tablets and even i-watch. This field is in continuous development and every newly released generation is opening new possibilities to the engagement with the user's context and increase security threats. The European Union Agency for Network and Information Security (**?**) listed in a survey the top ten security information risks for smart phone users. The number one was data leakage resulting from device loss or theft. This result was also featured by the US-CERT (United States

Computer Emergency Readiness Team) which also mentioned that the number of new vulnerabilities has jumped 42% from 2009 to 2010.
In order to fight that, smart devices are usually equipped with three authentication factors: something you know, something you have, and something you are. What you know comes as the main security recommendation for any user that is to set up his phone with a pin or a strong password. But even that level of security can be overcome if an attacker has enough time and access to the device. From the user's perspective, that type of authentication has a very low usability therefore a user might choose to store his password on the device for easier access and by that compromise its security. Something you have is by proving possession of something external to the system. Common choices for proving possession are: hardware tokens that generate one-time passwords, access to an e-mail address, the mobile device itself can be registered with an application, and then, possession of the device can be used as a something you have authentication factor. Choices for something you know that require a user carry an additional device are less convenient for the user. One of the reasons for the popularity of mobile devices is convenience. The something you are factor uses biometrics to authenticate users. Biometric based techniques are multiple such as keystroke analysis that was discussed in a research published in the International Journal of Information Security in 2007 (**?**). This paper identified two typical handset interactions, entering telephone numbers and typing text messages. It was found that neural network classifiers were able to perform classification with average equal error rates of 12.8%. Based upon these results, the paper concludes by proposing a flexible and robust framework to permit the continuous and transparent authentication of the user, thereby maximizing security and minimizing user inconvenience, to service the needs of the insecure and evermore functional mobile handset. Also, in 2009, a paper was published discussing a different form of keystroke dynamics with the finger pressure (**?**). The finding has shown that, the finger pressure gives the discriminative information more than keystroke dynamics with the k-NN analytical method. Moreover, using only the finger pressure produces high accuracy rate of 99%.
Combining multiple biometrics may enhance the performance of personal authentication system in accuracy and reliability. In Combining fingerprint and voice print biometrics for identity verification: an experimental comparison (**?**), 13 combination methods were compared in the context of combining the voice print and fingerprint recognition system in two different modes: verification and identification. The experimental results show that Support Vector Machine and the Dempster-Shafer method are superior to other schemes.
These authentication methods have proven their weakness in terms of usability and also efficiency. These methods are represented in the phones in the form of different screen lock mechanisms. From these mechanisms, we can name a few, such as:

– A simple swipe which does not provide security at all and is simply used as a screen saver.

– Face unlock where the user provides a shot of his face that is then recognized by the device and used to unlock it. This method has proven its weakness and its incapability of recognizing the user if the surrounding conditions of light mainly do not match the ones on the day he saved the settings.
– Face unlock and voice which combines the facial with the voice recognition. If the user is found in a place where he cannot raise his voice to the same pitch as the one used when he set up this security then the authentication will fail.
– Pattern which is the most common form of authentication and yet still weak since an adversary can guess the pattern of the user by simply checking the screen of the phone in an appropriate angle to see traces of the finger.
– PIN and password which are considered as a medium to high security is a combination of numbers or characters chosen by the user and required to be entered at every attempt to unlock the screen which can become quite annoying.

The above mentioned methods are becoming more and more annoying for the user since he has to repeat the same action multiple times a day often over 100 times. These types of authentication are user dependent and require his immediate intervention and input in order to proceed. And by that, any explicit action can be memorized by an adversary and used to unlock the device without the owner's consent. Also, once the device is unlocked, the security feature is deactivated even if it was not with its rightful owner. Therefore, an additional layer of security is required, one that does not require direct user intervention, but works implicitly and continuously to decide whether the user is indeed the authorized one. The proposed system aims at reducing the number of explicit authentication. Its purpose is not to replace the common authentication methods, but rather to complement them. That is, the user can still use his chosen authentication method, but once the phone is unlocked, the implicit authentication takes charge to determine if the user is indeed the owner or an attacker.
In order to be able to decide that, the device has to gather user centric data that will uniquely characterize the owner. As an example of such data is the gestural input. In the paper Biometric-rich gestures: a novel approach to authentication on multi-touch devices (**?**), a comprehensive set of five-finger touch gestures was defined, based upon classifying movement characteristics of the center of the palm and fingertips, and tested in a user study combining biometric data collection with usability questions. Using pattern recognition techniques, a classifier was built to recognize unique biometric gesture characteristics of an individual. 90% accuracy rate was achieved with single gestures, and significant improvement noticed when multiple gestures were performed in sequence. User ratings aligned well with gestural security, in contrast to typical text-based passwords.
Another implicit authentication technique discussed in "Implicit user re authentication for mobile devices" (**?**) included the observation of user-specific patterns in file system activity and network access to build models of normal behavior. The proposed system was able to distinguish between normal use and attack with an accuracy of approximately 90% every 5 minutes and consumed less than

12% of a typical laptop battery in 24 hours.

The main focus of our study is to extract the user behavior and transform it into a biometric signature that can be used to authenticate the user. We will attempt to discover whether it is possible to extract unique user signature from the user behavioral pattern to be used as implicit authentication mechanism. What kind of user centric information (and in what frequency) should be collected in order to detect the user behavioral pattern? How to transform the detected pattern into a unique signature? What correlation methodology should be used to verify the extracted signature?
In this work, we lay foundational work for implicit authentication through the capture of a user's unique behavioral pattern. The proposed system aims at reducing the number of explicit authentication. Its purpose is not to replace the common authentication methods, but rather to complement them. That is, the user can still use his chosen authentication method, but once the phone is unlocked, the implicit authentication takes charge to determine if the user is indeed the owner or an attacker. To achieve this, we introduce a technique by which we capture the signature of the application usage of a user. First, we collect application related data and in particular the duration of use. Next, we use a mathematical algorithm that will convert that data into a function particular to this user. This function will be used at run-time to determine if the user is indeed the rightful owner or an attacker. Our findings support that this is an approach with great potential. Thus, the main contribution of this work is a framework that helps us understand the user behavior and transform it into a unique signature that can be used to authenticate the user. The study provides an insight into quantifying user behavior and using it as a comparison standard. The remaining parts of this report are organized as follows: Chapter II introduces the related work. Chapter III presents the architecture, the different components of MUSEP and the behavioral pattern extraction. Chapter IV goes through the first attempt to select an appropriate mathematical model with the use of the Discrete Fourier Transform. Chapter V details the work done using the cubic spline interpolation along with the experimental setup and results of evaluating this method. Finally chapter VI, gives an overview about future work.

2 Related work

Implicit authentication is very broad topic and has been discussed by multiple papers. We will first look into the phone recognition, next we will go through some research concerning the user recognition. These researches are divided between looking into the behavioral pattern of the user, the keystroke analysis, and the gait recognition.

2.1 Phone recognition

When discussing a pattern of usage, the user is the first thing that comes to mind. However, the phone itself can present a pattern of usage that would make it detectable. The paper "Who do you sync you are? Smartphone Fingerprinting via Application Behavior" (**?**) tackles that subject in particular. The research looks into the timing and data volume of network traffic generated by a device. They relied on traffic generated by applications such as Facebook, WhatsApp, Skype, Dropbox, and others. For each packet generated by these applications, they recorded the arrival time, the size of the packet, and the direction whether incoming or outgoing packet. Also, they analyzed the burst which represent the peak of data transfers from the same type of connection, for example TCP packets. By using the K-NN classifier, they extracted what they called "fingerprint" of the phone. Following multiple experiments, they concluded that in about 15 minutes, the phone can be recognized with more than 90% accuracy rate.

2.2 Authentication mechanisms controlled by the phone

Today's mobile devices are equipped with multiple sensors making them prone to attacks. The researches in the past decade have been guided towards improving their security measures and authentication mechanisms. In order to be considered as "smart" device, Fisher et al. (**?**) debate in their paper "Smartphones: Not smart enough?" the idea that a phone should be able to scale up or down its authentication mechanisms based on contextual information received from the device sensors. And by that, the phone would be able to assess the risk and match the corresponding authentication mechanism. First, the paper defines high and low risk scenario where the high risk represents the public use of credit card information and the low risk such as saving passwords onto personal devices in order not to enter them at each sign in. Next, they describe four device context with examples on how the device should behave in low and high risk scenarios. For example, the device unlock is a common procedure available in all smart phones. After unlocking our device, we have access to all personal information, except those protected by an extra layer of password security. In a high risk scenario, the context-aware device should at first sense that the user picked up the phone and is moving it towards his face. Then, it should turn on the camera and scan his face for facial recognition to confirm that it is indeed the owner. Next, it should scan for any know Wi-Fi or Bluetooth devices nearby to determine the user's location and assess using the microphone also, if the user is in a crowded space. In a low risk scenario, the phone would just unlock once it recognizes the user's face. Anyone who attempts to unlock to device other than the legitimate user, would have his photo taken and saved within the device. The collection of such data would raise privacy concerns, therefore, the future work will look into minimizing the amount of data collected and aggregate any stored data. Also, they will attempt to understand how mobile device users construct a mental threat model in a variety of contexts and incorporate physical world

factors into contextual threat models.

2.3 Behavioral pattern

User implicit authentication can be achieved by looking into the behavioral pattern of the user. In 2009, in the Palo Alto research center, a paper was published on this same topic. This research (**?**) introduces the notion of implicit authentication, the ability to authenticate a user to its device based on common actions that the user performs. This paper focuses on the use of this type of authentication for Mobile Internet Devices in particular. Not omitting the fact that implicit authentication can be used in a multitude of other fields such as computers, medical devices to help preserve patients medical records, military equipment and out-of-band transaction verification. This paper evaluates a technique to compute and maintain an authentication score based on recent activities of the user. The scoring varies depending on a set of positive and negative events and depending on time elapsed. A positive event is defined as a common habit of the user, and when that occurs, the score increases. A negative event is a non-common event for the user, when that occurs, the score decreases. Time elapsing decreases the score if during that time, the user has usually high activity. When the score goes below the event-specific threshold, explicit authentication is needed by the user in order to access that feature of his device. The different data sources that can be used to make authentication decisions are grouped into 3 types: device data, carrier data and third party data. The device data is any data provided by the phone itself such as GPS coordinates, WiFi/Bluetooth connectivity, application usage, biometric-style measurements such as keyboard typing pattern and voice data. The carrier data can be used to know the user's approximate location and phone call patterns. The third party data such as cloud services can also be used since an increasing number of applications are hosted online. The architecture of the implicit authentication model will be as follows: past behavior will be the key for the learning algorithm, then based on the user model, and recent user behavior, a scoring algorithm will compute a final score based on which it will be decided whether the user is the original device owner. User modeling assumed in this paper is using independent features, where for example, a user's location is independent from its phone call log and any other activity. The data collected to perform this experiment consisted of emails, calls, SMSs, location, contacts, calendar, tasks, memos, alerts, battery level, (un)holstering, USB connections, power on/off, SD card removal/insertions. This data was from a blackberry device, over the period of 3 months. In order to simplify the research, the analysis was done on phone data and location data. Phone data in particular was analyzed based on the lapse of time since previous call, as for location data, they used the interactive clustering algorithm to compute clusters of the most frequently visited locations. The scoring algorithm was applied on this collected data and noticed that the score decreases to zero during the periods knows as active, and during that specific day, were not. Another experiment was conducted where an adversary calls a set of unknown numbers from the user's device, and the score

also quickly decreased to zero. As future work, they will attempt to make use of all features for the scoring, and report results on false positive and false negative rates, research methods to model the dependence between different features (i.e., activities) and research methods to model adversarial behavior.
SenSec (**?**) is an application prototype that constantly collects sensory data from accelerometers, gyroscopes and magnetometers and constructs the gesture model of how a user uses the device. SenSec calculates the sureness that the mobile device is being used by its owner. Based on the sureness score, mobile devices can dynamically request the user to provide active authentication (such as a strong password), or disable certain features of the mobile devices to protect user's privacy and information security. The experiment started with offline user classification by asking a set of 20 random volunteer to repeat 5 to 10 times a certain set of actions, pick up the phone, unlock it, open the email application, lock the phone and return it to the table. The online user authentication consisted of giving a phone for users for 24 hours with the SenSec application running on these phones. A sureness score is calculated. If it falls below a preset threshold while certain operation is performed, an authentication screen will be pop up asking user to enter a passcode. Next these same phones are given to other participants as a negative testing stage. As result, user studies show that SenSec can achieve 75% accuracy in identifying the users and 71.3% accuracy in detecting the non-owners with only 13.1% false alarms. Also, SenSec bears an average 4.96 seconds detection delay.

2.4 Keystroke analysis

Touch me once and I know it's you! Implicit Authentication based on Touch Screen Patterns (**?**) paper introduced the idea of a second authentication level. That is, if an attacker has already breached the first level of security, in this case, a lock pattern, the implicit authentication should be able to figure out that the user is an intruder. In order to perform that, the paper suggests to look into the way the user performs the input given the assumption that the intruder already in possession of the user's password pattern. The experiment designed in order to test this idea started by collecting data from 48 users on 4 different locking patterns (horizontal, vertical, vertical with two fingers, diagonal). The data collected was analyzed using dynamic time warping (DTW). This algorithm looks for similarities between sets of data and calculates the cost to match one onto the other. The result is a warp distance that can be used to determine how similar a set is to a reference set. In this work, a sequence consists of a time series of touch screen data (all combinations of X-coordinate(s), Y-coordinate(s), pressure, size, time). The reference set is the one used to identify the owner of the device as a signature of that owner. For each unlock screen, the reference set was created by taking the first 20 unlocks (each one a single unlock) for each user. This first round of testing showed some very low accuracy levels. In the best case, the true negative rate was 57%. This means that a little bit more than four out of ten attacks would have been successful. This was strongly influenced by

the time duration of the tests, the environment which was not realistic, and the fact that the participant were informed on how to act with their devices. In the second part of the paper, a more realistic approach was taken for the test. An android application was developed and sent to the participants by email along with a specific pattern that was assigned randomly. For instance, out of the 26 participants, for whom valid attacks existed, six reached an accuracy of 90% or higher. This second approach increased the overall accuracy by more than 20%. Overall, it can be stated that using touch screen data to identify users works to a certain degree. This is supported by the fact that increasing the threshold for valid authentication attempts improves overall accuracy. As future work, they attempt to improve accuracy of the results, also they will be implementing a prototype based on the presented approach that does the calculation on the mobile device to perform another long-term study based on this application.

Bo, Zhang et Al. feature in their study a framework entitled SilentSense (**?**). It consists of tracking the touch actions of the user and combine them with a movement based biometrics in order to verify whether the current user is the owner or guest/attacker. This approach showed that the user can be identified with an accuracy over 99%. For one operation on the device, the framework could capture multiple information, including: the coordinate on the screen of both touch down and release; the duration of one interaction; the sensory data from both accelerometer and gyroscope, the pressure for the finger touching on the screen, and the motion condition of the user. This detection combination was tested in a static and dynamic scenario. In the first, they evaluated the performance through three different applications, including Message, Album, and Twitter. It was noticed that the framework could reach over 80% accuracy within ten event observations, and the owner will be judged within 6 observations. As for the dynamic scenario, the framework collected their processed vertical and horizontal accelerations in the earth coordinate system and combined them with touch event features. After 12 steps, the accuracy to identify a guest can achieve 100% and after 7 steps, the accuracy to identify the owner can achieve 100.

Dividing that kind of data by application seemed to improve accuracy of the results. Looking at the application alone, it contains user centric data more than the phone itself. The application "knows best on when to authenticate and how to authenticate" (**?**). In this research, the application developer decides a suitable classifier depending on the type of application. For example, for a browser, a classifier based on touch input behavior would provide more accuracy than one with keystrokes data. This application centric approach achieved over 85% accuracy rate after 50 training samples.

Classifying movement characteristics of the center of the palm and the fingertips was considered among the promising authentication techniques (**?**). The five-finger touch gestures achieved a 90% accuracy rate in recognizing an owner based on pattern recognition techniques.

Frank, Biedert et Al. propose a framework Touchalytics (**?**) that relies on touchscreen input as data source. They discussed in their paper the ability to continuously authenticate users based on the way they interact with the touchscreen of

a smart phone. That interaction is typically the way the user scrolls text on his phone. It includes sliding horizontally over the screen and sliding vertically over the screen to move screen content up or down. This behavior covers browsing through images or navigating to next screens, or reading emails or documents or browsing menus. Every user interacts differently with his phone in this context and can by that be authenticated according to this particular feature. In order to be able to distinguish between different users, the paper suggests the usage of two different classifiers k-nearest-neighbors (kNN) and a support vector machine with an rbf-kernel (SVM). The kNN classifier takes every new observation (here: a stroke) and locates it in feature space with respect to all training observations. The classifier identifies the k training observations that are closest to the new observation. Then, it selects the label that the majority of the k closest training observations have. SVM generalizes from the observed data, i.e., it forgets the individual observations after training and only saves the decision. Experiments were conducted where a set of users are given a text to read on their phones and their stroke pattern was recorded. Overall, the authentication difficulty seems to increase with increasing temporal distance to the training phase. The individuals in the experiment would complain from having to read a long text and gave up half way. Interestingly, the long-term authentication of the scrolling classifiers is an exception as its median error rate is lower than for the inter-session authentication. Thereby, depending on the authentication scenario, there is approximately a 0% to 4% chance that the correct user will be rejected or that a false user will be accepted. For some scenarios, this error rate is still too high for the system being directly implemented as is. However, this result demonstrates that touch-based continuous authentication is feasible.
Itus (**?**) is an open-source framework that can be deployed off-the-shelf and that combines SilentSense and Touchalytics. It provides an application easy to adapt, extensible and with low performance overhead.

2.5 Gait recognition

Utilizing the physiological and behavioral biometrics along with environmental factors to recognize the owner of a device is one approach in implicit authentication. Assuming that every person has his own movement pattern, that is his manner of walking or moving his feet, then it can be used to authenticate that person. Mobile devices these days are equipped with gait and location sensors that allow them to track this movement pattern. Using correlation to model the data in order to identify the user turned out to be more performing than the FFT (Fast fourier transform) providing a 7% error ratio with 10% for FFT (**?**) (**?**). Also, the paper ePet: when cellular phone learns to recognize its owner (**?**) used that gait data and applied a different algorithm. Based on the fact that that data is a time series, they chose a variant of Dynamic Time Warping (DTW) algorithm called FastDTW. The purpose is to assume that the phone will attach to its owner so much that it will be able to distinguish whenever it is being carried by someone other than its owner and take security measure automatically.

Their future work included the actual implementation of the recognition system based on this technique.

3 MUSEP Architecture

The MUSEP system is composed of the three software components as depicted in figure **??**. The three components are the learning component, the mathematical model and the intrusion detection component. The MUSEP is executed in two main phases: the learning phase and the intrusion detection phase.

3.1 The learning phase

The learning phase comprises one main component; the learning component. The user behavioral pattern will serve as an input for the device. This data is collected via an application that works in the background continuously while the user is using his phone. The data in its raw form is filtered, preprocessed and stored within the device for later use.

3.2 The intrusion detection phase

The intrusion detection phase is composed of two main components; the mathematical component and the intrusion detection component. In the mathematical component, the data stored in the first phase will serve as a comparison tool for the decision phase. The input of this model is the start time of the activity (considered as abscissa x) collected at run time which is provided to the cubic spline function; the mathematical part of the overall algorithm. Next, a decision making tool will use both the data stored in the device, and the result of the cubic spline function in order to come up with the proper decision. The mathematical model component (the cubic spline function) is used in collaboration with the intrusion detection component to form the intrusion detection phase. In the later component, the algorithm will compare the ordinate "y" from the stored data with the result of the cubic spline. The stored data will also serve to determine a threshold (standard deviation) by which the decision of owner or adversary will be taken depending on the difference between the results.

In the next paragraphs, each part of the system and its different components will be detailed and explained.

3.3 User Behavioral Pattern

User behavior is defined as all kind of user interaction with his phone. That is, not limited to, the applications he uses, the time he uses them, the duration of use, and the order of use. This paper focuses on deriving a user pattern from that data. It is assumed that each user has his own habits of phone usage that distinguish him from another and these habits obey some functions and might

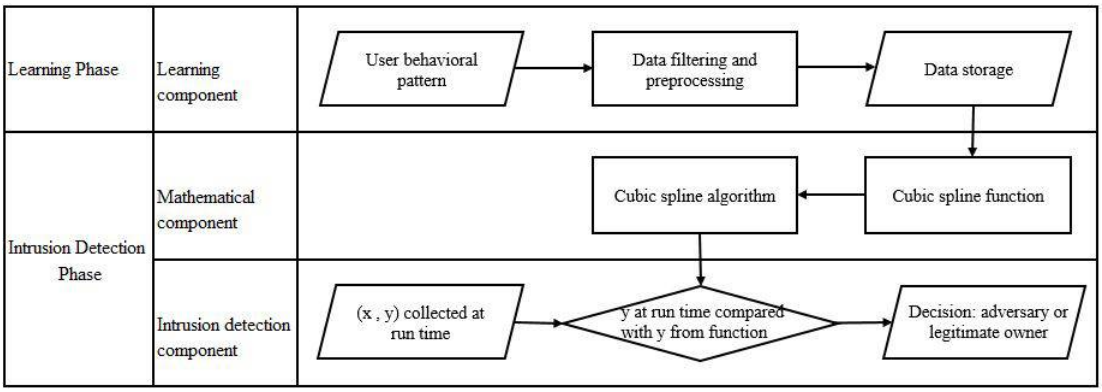

Fig. 1. MUSEP Architecture

show a periodic behavior. This uniqueness can be translated in the form of a user signature, which can also be used to recognize the owner of the device. In order to conduct this research, we restrict our study to smart phones with android platform. The first part of study consists of collecting user centric data to capture the user behavior. That data is collected using an android application that records user activity shown in figure **??**. This application was distributed to 10 users and data was collected over a sequence of 30 days. The users were asked to run the application on their phone and not to stop it till the end of the experiment. No special behavioral requirement was asked of them. Once

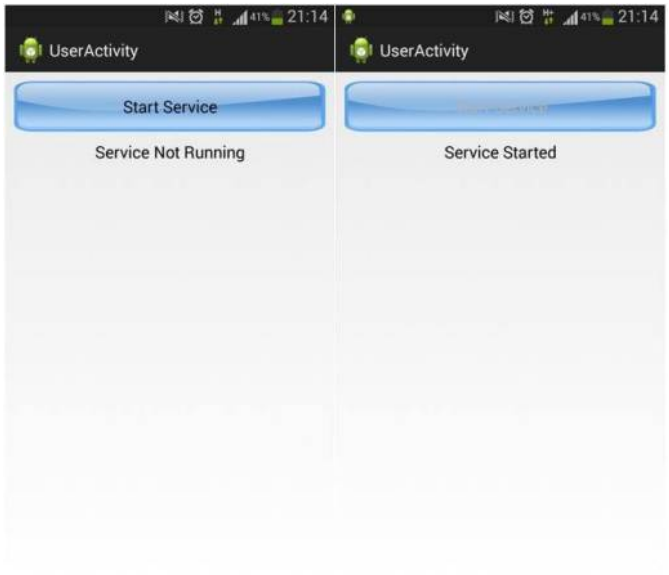

Fig. 2. User activity application

the experiment is completed, the user is requested to stop the service and click on the menu button to export the data collected by the application. A comma separated values (CSV) file shown in table **??** is generated and it contains the following data:

1. ID of the action which is a serial number that starts at 1 and is increased by 1 for each new registered action
2. Start time which refers to the date and time of the action
3. Start time sequence
4. End time which refers to the date and time of the action
5. Application name that is used as given by android
6. Application ID a number given to each application starting by order of use
7. Duration of use of the application in seconds

This data can be used for multiple interpretations such as looking into the order in which the user is using his applications, the duration of no-activity between applications, the duration of use of each application. In this research, we focus our study at analyzing the pattern in the duration of use of each application. As an example, figure **??** shows the duration of use of the Whatsapp application plotted against the start time of each usage of that application during 5 full weekdays for two users. The data is taken raw and not manipulated in any form. In our study, we assume that each user presents a unique pattern in the duration of use of each application for the same time frame. We will attempt to prove this hypothesis using real collected data and with the support of a mathematical model.

Table 1. Data as extracted from the User Activity application

id	Start Time	Start Time Sec	End Time	Application Name	Application ID	Duration
10	10/29/2014 10:13:46 AM	36826	10/29/2014 10:13:53 AM	com.android .email	2	6
11	10/29/2014 10:13:53 AM	36833	10/29/2014 10:14:09 AM	com.whatsapp	3	16
12	10/29/2014 10:14:09 AM	36849	10/29/2014 10:14:11 AM	screen off	4	1
13	10/29/2014 10:14:11 AM	36851	10/29/2014 10:14:53 AM	com.whatsapp	3	41

As an example of raw collected data, graphs in figure **??** represent the duration of "Whatsapp" application usage in seconds for the same time frame for 2 different users. The graphs are at the same scale and represent data taken at the same time. We can already notice the difference between the users and a sort of periodicity in the behavior of the same person simply by examining the graphs. Next, we needed to sample the data for proper analyzing. As a first test, and

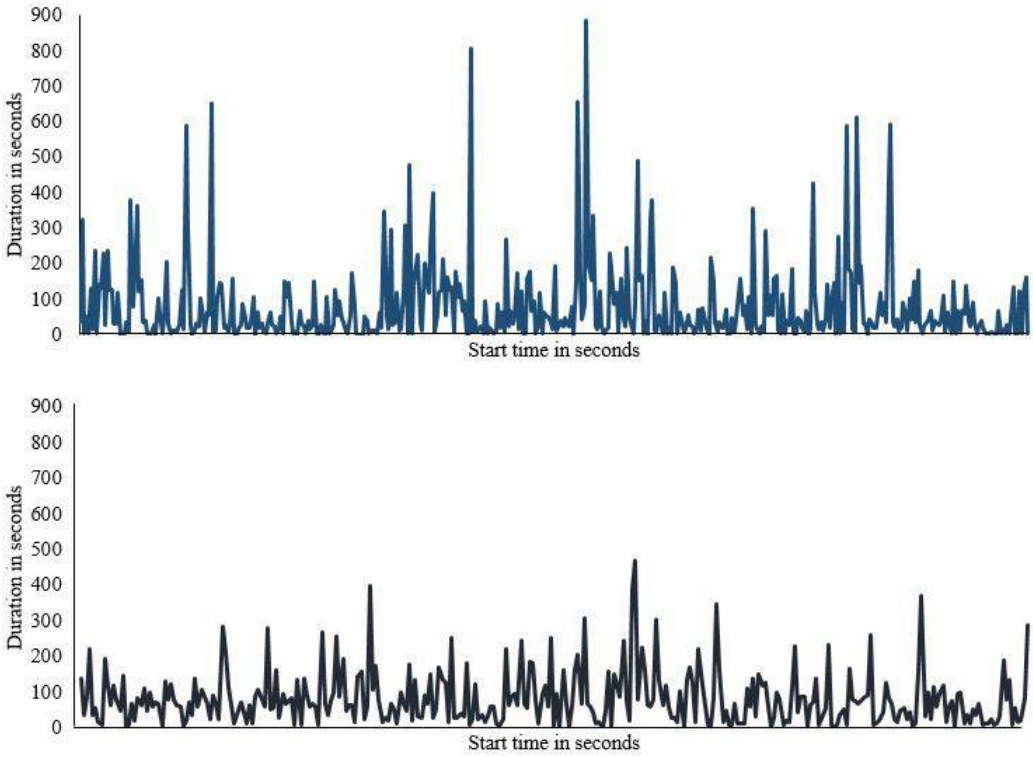

Fig. 3. Whatsapp duration vs start time for user A and user B

after examining the original data and the users' usual working hours, we decided to divide the data into 5 time slots according to the hours of a day:

1. Time Slot A: 00:00 AM - 06:00 AM
2. Time Slot B: 06:00 AM - 12:00 PM
3. Time Slot C: 12:00 PM - 17:00 PM
4. Time Slot D: 17:00 PM - 20:00 PM
5. Time Slot E: 20:00 PM - 00:00 AM

Also, for proper modeling, data is normalized; start time is considered as 0 for the first time recorded in the time slot, then the interval between 2 consecutive usages is added. Example:

The graph in figure **??** shows the data for one application (Whatsapp), one user, and one time slot over 5 consecutive weekdays. We can already notice that the data collected in the time slot is too large to be properly modelled and might not generate highly accurate results. Therefore, we decided to alter the way we are selecting the data.

Table 2. Start time normalization

Start time	Start time normalized
1/7/2015 18:00	0
1/7/2015 18:07	18:07 - 18:00 = 7
1/7/2015 18:16	18:16 - 18:07 = 16

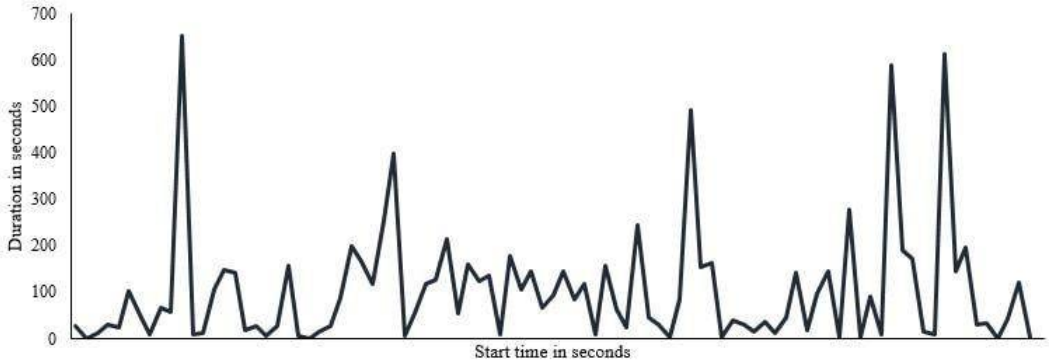

Fig. 4. Whatsapp usage over 1 time slot, over 5 days

3.4 Second Approach

Instead of looking at the data at that large scale, we decided to reduce the time scale by looking at individual hours over several days. That is, for example, examining the behavior of a user for one application, at 6:00 PM over seven (7) weekdays. Examining the data in this form would create a much more consistent pattern than when looking at it as a whole.

1. Data filtering steps
   First, the time stamp is divided into hour, minute and seconds. Then a new calculated member is created to convert the minutes and seconds into seconds to retrieve the start time in terms of seconds within that hour (figure **??**). Next, the data is filtered by application (Whatsapp in our example in figure **??**). Then, data is divided weekday and weekend, then gathered for the same hour for the same application over the duration of the data collection. In our example, data is taken in January 2015 with 10,15 and 17,18 as weekends. Next, duration is filtered to values between 5 and 180 seconds in order to remove the readings that were not meaningful in our approach (figure **??**).
   The data is now ready for analysis, as an example, we will take the hour 18:00. The collected data set consists now of the converted start time in seconds, and duration of use in seconds. The columns start time, application name, end time are no longer needed. That data set is ordered in ascending converted start time.
   The "sec converted" column can be considered as the abscissa, and the

Seq	Start_Time	Application_Name	Duration	HOUR	MINUTE	SECOND	SEC CONVERTED
1	1/7/2015 0:00	com.whatsapp	22	=HOUR(B2)	=MINUTE(B2)	=SECOND(B2)	=H2*60+I2
2	1/7/2015 0:01	screen off	0	0	1	2	62
3	1/7/2015 0:01	screen off	18	0	1	21	81
4	1/7/2015 0:02	com.whatsapp	28	0	2	46	166
5	1/7/2015 0:03	screen off	2220	0	3	16	196
8	1/7/2015 0:40	screen off	1116	0	40	42	2442
9	1/7/2015 0:59	screen off	3959	0	59	39	3579
12	1/7/2015 2:06	screen off	18362	2	6	41	401
31	1/7/2015 7:12	screen off	1034	7	12	48	768
37	1/7/2015 7:31	screen off	199	7	31	41	1901
43	1/7/2015 7:36	screen off	230	7	36	11	2171

Fig. 5. Step 1: Time conversion

Seq	Start_Time	Application_Name	Duration	HOUR	MINUTE	SECOND	SEC CONVERTED
1	1/7/2015 0:00	com.whatsapp	22	=HOUR(B2)	=MINUTE(B2)	=SECOND(B2)	=H2*60+I2
4	1/7/2015 0:02	com.whatsapp	28	0	2	46	166
62	1/7/2015 7:45	com.whatsapp	19	7	45	10	2710
64	1/7/2015 7:46	com.whatsapp	37	7	46	44	2804
85	1/7/2015 8:01	com.whatsapp	4	8	1	1	61
124	1/7/2015 8:53	com.whatsapp	146	8	53	13	3193
135	1/7/2015 9:00	com.whatsapp	2	9	0	17	17
136	1/7/2015 9:00	com.whatsapp	3	9	0	21	21
162	1/7/2015 9:25	com.whatsapp	44	9	25	46	1546
188	1/7/2015 9:41	com.whatsapp	156	9	41	57	2517
201	1/7/2015 9:48	com.whatsapp	185	9	48	9	2889
212	1/7/2015 9:51	com.whatsapp	88	9	51	48	3108

Fig. 6. Step 2: Application filtering

"Duration" its ordinate (figure **??**).

2. Data preprocessing
   To analyse this data set and avoid fluctuation and negative values in the interpolation, data is sampled at a rate of 8.33 x 10^{-3} Hz, that is a reading every 2 minutes. Since the user does not necessary use any application at that particular rate, the data is distributed to 31 points by assigning it to the higher start time. The example is shown in tables **??** and **??**. The first point is (0, 0). The 2nd start time 64 is less than 120, therefore, the duration 48 is assigned to 120. As for the start time 275, 287, 340, they all fall below 360, so an average of their duration is taken and assigned to 360. As a result: Sometimes, the data acquired does not fill the 31 points that represent an hour. As a solution, midpoints are used to bridge gaps. Using this method, the same sample data used earlier is filtered, and the result is a curve showing one application (Whatsapp), one user, and one hour over 5 consecutive weekdays (figure **??**). We can notice that this time, the data is less and can be modelled. We will be looking next at a way to quantify that behavior using a mathematical function.

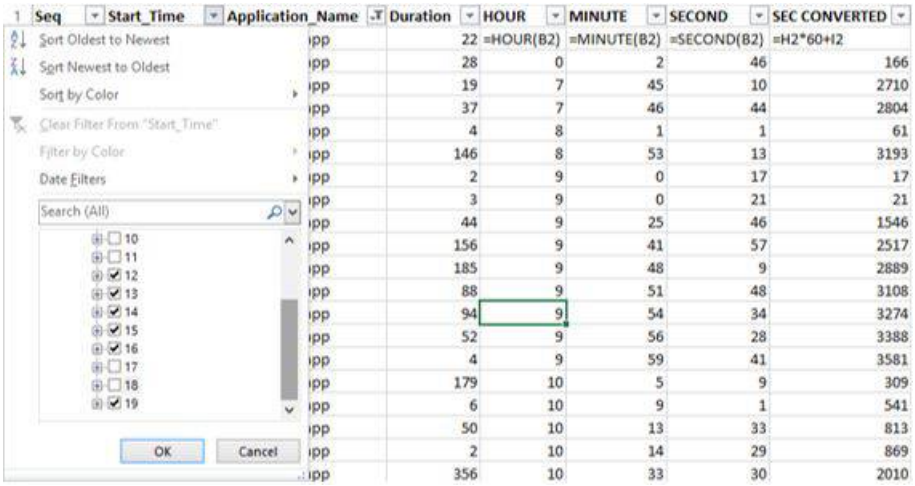

Fig. 7. Step 3: Weekday selection

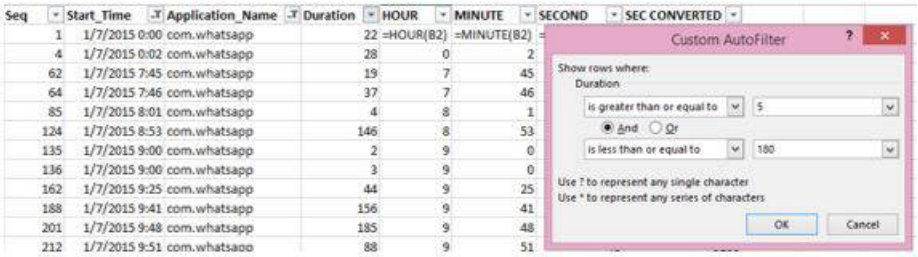

Fig. 8. Step 4: Filtering for duration between 5 and 180 seconds

Table 3. Original data

Original start time	Duration of use
64	48
172	59
203	58
275	5
287	5
340	42
414	7
461	39

4 Cubic spline interpolation

The data collected from the user activity application is filtered and ready for modelling. Since we have a set of tuples (x , y), a polynomial function is needed.

SEC CONVERTED	Duration
48	62
91	20
329	45
543	28
579	27
844	8
1167	6
1200	84
1409	12
1509	47
1512	52
1542	93
1654	68
1702	7
1748	116
1748	34
1953	13
2143	16
2173	68

Fig. 9. Result of the filtering

Table 4. Original data reallocated

Reading every 2 minutes	Allocated duration
0	0
120	48
240	59
360	17
480	23
600	21
720	40
840	33

As a first test, on one time slot, one application was chosen. If this data were modeled using high degree polynomial, the result would be as shown in figure **??**, the curve would jump to high results at undesired locations. Also, the curve does not respect the points given to it and is far from being accurate. The plot below was conducted using a 9th degree polynomial using Matlab (**?**).

Given the low accuracy rate with a regular polynomial, we needed a function that would reflect the actual user behavior without compromising its integrity. Using the cubic spline polynomial leads us to our exact goal by modelling the dataset without an error threshold. The reason for that is that this function is based on individual cubic polynomials that link each 2 points in order to create a smooth curve that passes through all the points. In the plot in figure **??**, we can see the same set of points modelled using the cubic spline function.

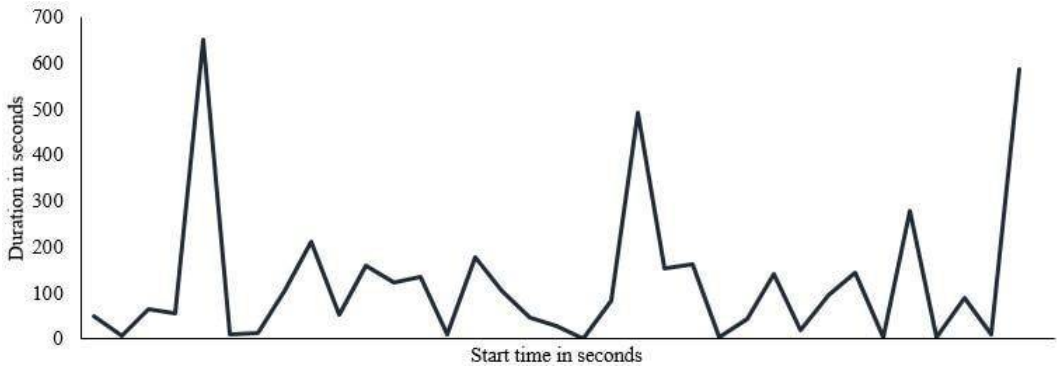

Fig. 10. Whatsapp usage over 1 hour over 5 days

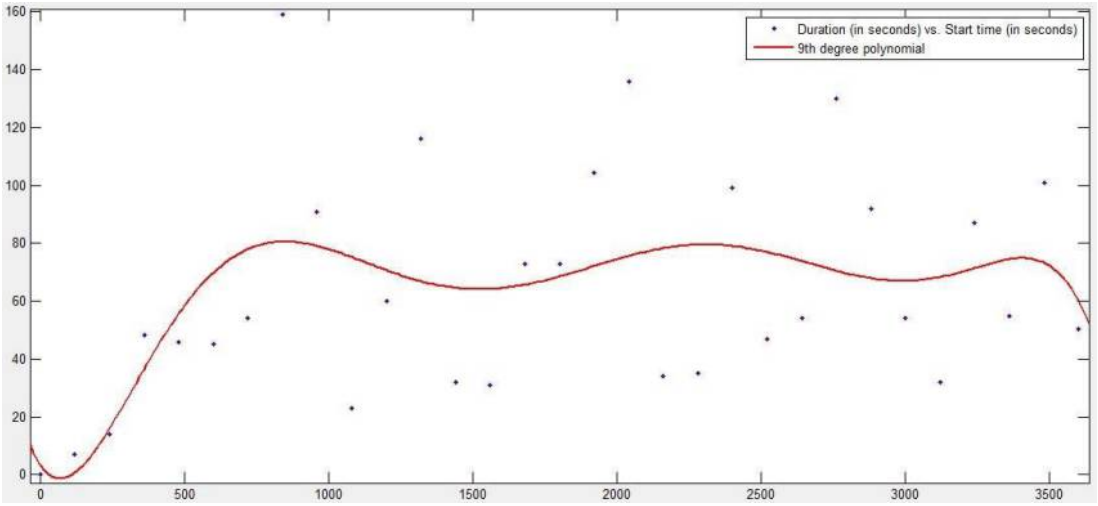

Fig. 11. Data modelled using 9th degree polynomial

4.1 Modelling data set

The cubic spline interpolation is used to model the dataset as shown in the first graph in figure **??**. The graphs show the duration of usage of the same application in seconds, for the same hour, the same dates taken for 4 different users against the 31 points that represent the start time in seconds. From the shape of the function, we can start to notice the de-correlation between users. This function will be later used to recognize the user from the duration of use of an application and from the pattern of use that the function has learned throughout the first phase. At run time, the phone can send to the function the start time of an application and it will return the expected duration of use. Comparing the obtained value with the original run-time value can provide information about the authenticity of the user.

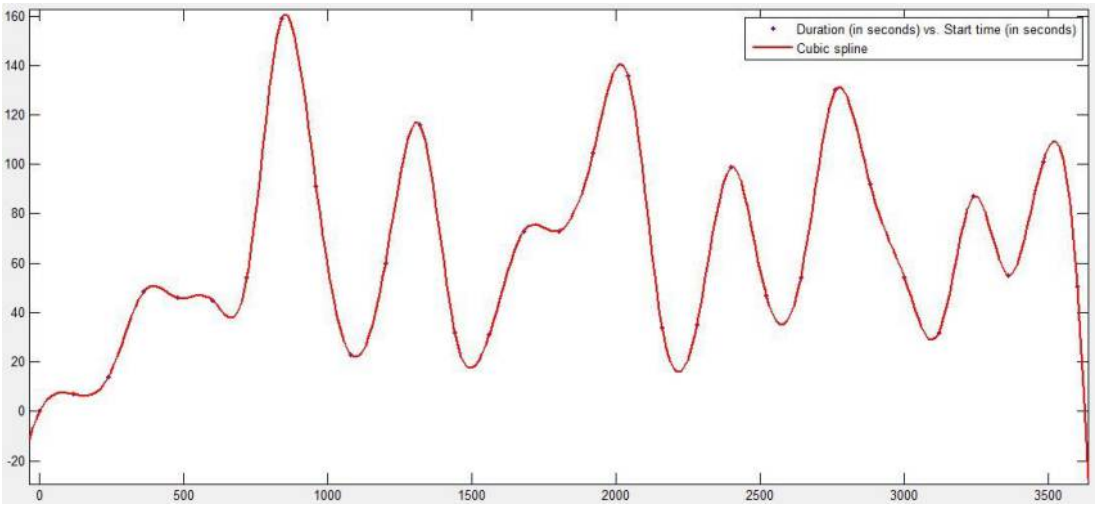

Fig. 12. Data modelled using cubic spline

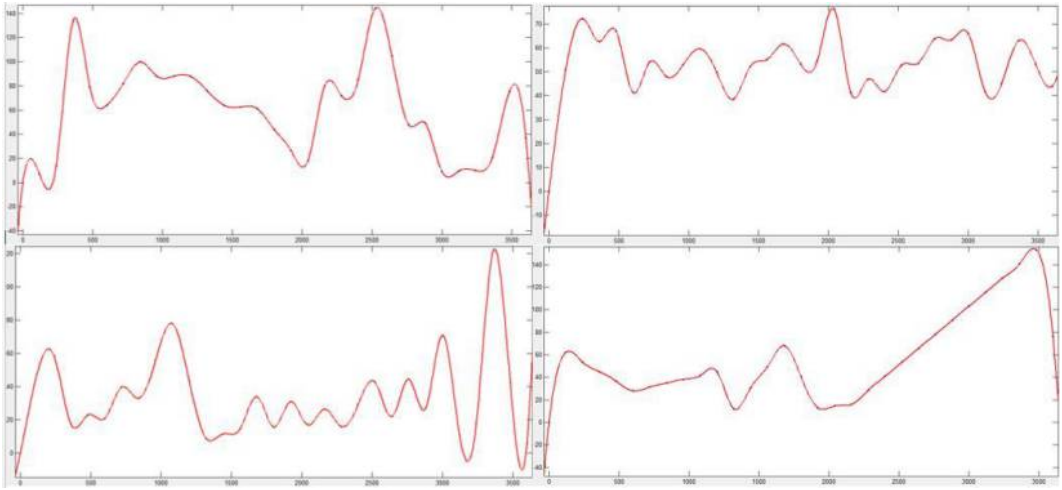

Fig. 13. Data for four users interpolated using cubic spline polynomial

4.2 Error Threshold

To further prove the independence of the users, additional tests were carried out. The value of x (start time) was taken from one user A and tried within another user's function (user B). The function generated the expected duration of use of user B (with the data of user A) which was compared with the original duration of use from user A for the same time slot and same application. Then, the difference between the resulted duration and the original value was calculated. It is assumed that the larger the difference is, the less a correlation exist between

USER A PROCESSED DATA	
X	Y
-	-
120	41
240	43
360	45
480	37
600	28
720	20
840	12
960	8
1,080	27
1,200	45
1,320	12

USER B ORIGINAL DATA	
X	Y
64	48
172	59
679	82
1628	177
1912	44
2343	57
2868	32
2900	14
2910	128
3073	11
3439	53
3445	68

Fig. 14. User A processed data to be used to create the function, and User B original extracted data

Xi	f(Xi)
64	30.0029
172	43.4897
679	22.7379
1628	68.3334
1912	14.549
2343	34.2885
2868	25.0275
2900	29.3187
2910	30.6105
3073	51.1723
3439	126.273
3445	130.848

Fig. 15. User B values (x) tried within User A function

the users. That is, no two users can have the same function and by that present the same pattern of usage of an application. A sample of the differences between these two values are shown in figure **??** for one application for four users. The data of three users is used in the fourth user's function to generate the value of "y". It was noticed that if the users know each other and were using the same application to communicate with each other (such as phone, Whatsapp, Facebook messenger), the dependency was higher than when they did not. Also, the same experiment was conducted in order to compare data from a user within his own function. Data was taken from each user and tested within the function that was obtained from the data collected during the learning phase (after processing). In order to be able to compare, some days were excluded from the

Y USER B	Y RESULT FROM USER A FUNCTION	DIFFERENCE
48	30	**18**
59	43.48	**15.52**
82	22.73	**59.27**
177	68.33	**108.67**
44	14.54	**29.46**
57	34.28	**22.72**
32	25.02	**6.98**
14	29.31	**15.31**
128	30.61	**97.39**
11	51.17	**40.17**
53	126.27	**73.27**
68	130.84	**62.84**

Fig. 16. Differences between User B original values, and the result from User A function

User 2, 3 and 4 vs User 1			
User 2	User 3	User 4	Average
52	38	5	**32**
54	8	21	**28**
52	114	55	**74**
57	18	87	**54**
38	13	129	**60**

User 1, 3 and 4 vs User 2			
User 1	User 3	User 4	Average
29	18	29	**25**
36	59	22	**39**
128	109	28	**88**
113	29	76	**73**
47	23	123	**64**

User 1, 2 and 3 vs User 4			
User 1	User 2	User 3	Average
111	55	41	**69**
28	12	53	**31**
19	42	43	**35**
22	68	119	**70**
54	31	57	**47**

User 1, 2 and 4 vs User 3			
User 1	User 2	User 4	Average
49	42	48	**46**
85	17	21	**41**
64	17	29	**37**
105	66	53	**75**
80	4	94	**59**

Fig. 17. Differences between original extracted values and plotted values

original data set and used in the function extracted from the remaining days. The results obtained were very interesting and the average error shows quite a difference from the previous experiment.

In figure **??**, the "Y ORIGINAL" column represents the actual duration collected at run time. The days from which this data has been extracted were excluded from the overall dataset that was processed to generate the user function. The start time collected at run time (column X) is used within the user's own function to generate the "Y FROM FUNCTION" value. By comparing the

X	Y ORIGINAL	Y FROM FUNCTION	DIFFERENCE
329	45	31.42	13.58
844	8	12.77	4.77
1512	52	62.39	10.39
2279	34	37.92	3.92
2774	31	8.82	22.18
2901	5	7.19	2.19

Fig. 18. Comparison of a user's data within his own function

original data with the result of the function, we can notice a very low average of error, that is 9.5 seconds in contrast with the high averages obtained in the previous experiment and which ranged from 25 to 88 seconds. Some more results with additional users can be seen in figure **??**. Since we have established that

	User 1	User 2	User 3	User 4
	30	5	21	10
	23	22	2	0
	31	14	5	19
	7	10	20	13
	7	4	9	22
Average	**19.6**	**11**	**11.4**	**12.8**

Fig. 19. Difference between original extracted values and plotted values for same user

users' data is not correlated and that two users cannot have the same pattern of usage, we can use the above results to set a threshold by user by application to base upon it our decision of owner or intruder. The error threshold is set based on the average of the differences within a user's own data obtained during the learning phase. This threshold, shown in table **??**, is set by application by user, for example, the above data is extracted for the Whatsapp application (not all data is shown).

5 Experimental Results

Using this approach to determine the legitimacy of the user of a device proved to be quite promising. It was tested using Matlab to simulate the actual environment. The results were divided between:

1. True positive: The user is indeed the owner of the device.

Table 5. Average of differences taken from full data of Whatsapp application

User	Average difference in seconds with other users	Average difference in seconds with user's own data
User 1	45	19
User 2	45	16
User 3	46	18
User 4	50	13
Average	46.5	16.5

2. True negative: The user is indeed an adversary.
3. False positive: The user is an adversary but the result suggests he is the owner.
4. False negative: The user is the owner of the device but the result suggests he is an adversary.

5.1 Experiments

In order to conduct the simulation of a true positive, a subset of the data was removed from each user consisting of a number of days, then the remaining data was modelled using the above mentioned method. The subset is later tested against the modelled dataset by providing it with the start time converted in seconds, and comparing the result obtained from the modelled dataset with the result from the raw data. If that result was below the threshold, then the user is indeed the owner and the result is counted as true positive. If the result is higher than the threshold, and since the user is the owner, the result is counted as false negative.
Next, in order to simulate the true negative, a dataset from other users was taken and tested within one user's modelled data. That simulates an attack on the device of the user by an adversary. If the obtained result was below the preset threshold, then it is counted as false positive. If the result was over the threshold, then it is counted as true negative.

5.2 Results

The results of the above experiments are depicted in table 8. We can notice that we were able to achieve a positive identification of the owner 70 out of 100 trials, and the intruder 7.6 out of 100 trials. These results are not as high as we would have expected them to be but they show a beginning of a promising idea. Also, these results are limited to a small subset of users and to one application, enlarging the point of view might enhance them greatly.

	AVERAGE
True Positive	70%
True negative	76%
False positive	24%
False negative	30%

Fig. 20. Results of MUSEP simulation

6 Future Work

The user behavior can be further expanded to cover things other than the application usage. Everything that is affected by the user can be regarded as user behavior, for instance, the speed of battery drain, the CPU percentage usage, data stream over the Wi-Fi and the mobile data network. In this research we have considered every application to generate a single user signature, what still can be explored is putting all collected user behaviors in a single matrix. Further, the matrix eigenvalues can be used as a unique signature. One can also analyze the sequence of applications usage and the interval between them. In this work, we have tried to identify and convert the user behavior into a unique signature, nevertheless, we do believe that there is still a lot to explore in this field.

References

Enisa.europa.eu. (2014), Top Ten Smartphone Risks ENISA, [ONLINE] Available at: https://www.enisa.europa.eu/activities/Resilience-andCIIP/critical-applications/smartphone-security-1/top-ten-risks. [Accessed 24 November 2014].

Clarke, N.L. and Furnell, S.M, (2007), Authenticating mobile phone users using keystroke analysis. International Journal of Information Security", 6(1), pp.1-14.

Saevanee, H. and Bhattarakosol, P., (2009), Authenticating user using keystroke dynamics and finger pressure, In 2009 6th IEEE Consumer Communications and Networking Conference (pp. 1-2). IEEE.

Wang, Y., Wang, Y. and Tan, T., (2004), Combining fingerprint and voiceprint biometrics for identity verification: an experimental comparison, In Biometric Authentication (pp. 663-670). Springer Berlin Heidelberg.

Sae-Bae, N., Ahmed, K., Isbister, K. and Memon, N., (2012), May. Biometric-rich gestures: a novel approach to authentication on multi-touch devices. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 977-986). ACM.

Yazji, S., Chen, X., Dick, R.P. and Scheuermann, P., (2009), July. Implicit user re-authentication for mobile devices. In International Conference on Ubiquitous Intelligence and Computing (pp. 325-339). Springer Berlin Heidelberg.

Mathews, J.H. and Fink, K.D., (1999), Numerical methods using MATLAB (Vol. 31). Upper Saddle River, NJ: Prentice hall.

Fischer, I.T., Kuo, C., Huang, L. and Frank, M., (2012), October. Short paper: Smartphones: Not smart enough?. In Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices (pp. 27-32). ACM.

Stber, T., Frank, M., Schmitt, J. and Martinovic, I., (2013), April. Who do you sync you are?: smartphone fingerprinting via application behaviour. In Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks (pp. 7-12). ACM.

Jakobsson, M., Shi, E., Golle, P. and Chow, R., (2009), August. Implicit authentication for mobile devices. In Proceedings of the 4th USENIX conference on Hot topics in security (pp. 9-9). USENIX Association.

Zhu, J., Wu, P., Wang, X. and Zhang, J., (2013), January. Sensec: Mobile security through passive sensing. In Computing, Networking and Communications (ICNC), 2013 International Conference on (pp. 1128-1133). IEEE.

De Luca, A., Hang, A., Brudy, F., Lindner, C. and Hussmann, H., (2012), May. Touch me once and i know it's you!: implicit authentication based on touch screen patterns. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 987-996). ACM.

Bo, C., Zhang, L., Li, X.Y., Huang, Q. and Wang, Y., (2013), September. Silentsense: silent user identification via touch and movement behavioral biometrics. In Proceedings of the 19th annual international conference on Mobile computing & networking (pp. 187-190). ACM.

Khan, H. and Hengartner, U., (2014), February. Towards application-centric implicit authentication on smartphones. In Proceedings of the 15th Workshop on Mobile Computing Systems and Applications (p. 10). ACM.

Sae-Bae, N., Ahmed, K., Isbister, K. and Memon, N., (2012), May. Biometric-rich gestures: a novel approach to authentication on multi-touch devices. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 977-986). ACM.

Frank, M., Biedert, R., Ma, E., Martinovic, I. and Song, D., (2013). Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. IEEE transactions on information forensics and security, 8(1), pp.136-148.

Khan, H., Atwater, A. and Hengartner, U., (2014), September. Itus: an implicit authentication framework for android. In Proceedings of the 20th annual international conference on Mobile computing and networking (pp. 507-518). ACM.

Shi, E., Niu, Y., Jakobsson, M. and Chow, R., (2010), October. Implicit authentication through learning user behavior. In International Conference on Information Security (pp. 99-113). Springer Berlin Heidelberg.

Feng, T., Liu, Z., Kwon, K.A., Shi, W., Carbunar, B., Jiang, Y. and Nguyen, N., (2012), November. Continuous mobile authentication using touchscreen ges-

tures. In Homeland Security (HST), (2012) IEEE Conference on Technologies for (pp. 451-456). IEEE.

Chiang, H.Y. and Chiasson, S., (2013), August. Improving user authentication on mobile devices: A touchscreen graphical password. In Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services (pp. 251-260). ACM.

Meng, Y., Wong, D.S. and Schlegel, R.,(2012), November. Touch gestures based biometric authentication scheme for touchscreen mobile phones. In International Conference on Information Security and Cryptology (pp. 331-350). Springer Berlin Heidelberg.

Shi, W., Yang, J., Jiang, Y., Yang, F. and Xiong, Y., (2011), October. Senguard: Passive user identification on smartphones using multiple sensors. In 2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) (pp. 141-148). IEEE.

Wolberg, G. and Alfy, I., (1999). Monotonic cubic spline interpolation. In Computer Graphics International, 1999. Proceedings (pp. 188-195). IEEE.

McKinley, S. and Levine, M., (1998) Cubic Spline Interpolation. College of the Redwoods.

Gafurov, D., Helkala, K. and Sndrol, T., (2006). Biometric gait authentication using accelerometer sensor. Journal of computers, 1(7), pp.51-59.

Tamviruzzaman, M., Ahamed, S.I., Hasan, C.S. and O'brien, C., (2009), November. ePet: when cellular phone learns to recognize its owner. In Proceedings of the 2nd ACM workshop on Assurable and usable security configuration (pp. 13-18). ACM.

Mantyjarvi, J., Lindholm, M., Vildjiounaite, E., Makela, S.M. and Ailisto, H.A., (2005), March. Identifying users of portable devices from gait pattern with accelerometers. In Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. (Vol. 2, pp. ii-973). IEEE.

Osgood, B., (2013). Lecture notes for EE 261: the Fourier transform and its applications. Stanford Engineering Everywhere.

Rao, K.R., Kim, D.N. and Hwang, J.J., (2011). Fast Fourier Transform-Algorithms and Applications. Springer Science & Business Media.

Klasson, K.T., (2008). Construction of spline functions in spreadsheets to smooth experimental data. Advances in Engineering Software, 39(5), pp.422-429.