# DEMOGRAPHIC GROUP PREDICTION BASED ON SMART DEVICE USER RECOGNITION GESTURES

Approved by:

Dr. Mitch A. Thornton - Committee Chairman

Dr. Sukumaran Nair

Dr. Delores Etter

Dr. Ted Manikas

Dr. Frank Coyle

# DEMOGRAPHIC GROUP PREDICTION
# BASED ON SMART DEVICE USER RECOGNITION GESTURES

A Dissertation Presented to the Graduate Faculty of the

Lyle School of Engineering

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Doctor of Philosophy

with a

Major in Computer Engineering

by

Adel R. Alharbi

(B.S., CSE, Qassim University, 2008)
(M.S., CSE, Southern Methodist University, 2012)

May XX, 2017

Alharbi , Adel R. B.S., CSE, Qassim University, 2008
M.S., CSE, Southern Methodist University, 2012

Demographic Group Prediction
Based on Smart Device User Recognition Gestures

Advisor: Dr. Mitch A. Thornton - Committee Chairman
Doctor of Philosophy degree conferred May XX, 2017
Dissertation completed May XX, 2017

Interacting with smart devices is a common experience and is becoming an integral part of daily life for many people. We hypothesize that a feature of smart device sensor data can provide biometric data that allows for classification of user demographics such as age, gender, and native language. In this work, we propose a demographic group prediction mechanism for smart device users based upon the recognition of user gestures. The core idea of our proposed approach is to utilize data from a variety of the internal environmental sensors in the device to predict useful demographics information. In order to achieve this objective, an application with several intuitive user interfaces was implemented and used to capture user data. The results presented here are based upon data collected from one hundred test subjects. These captured data are integrated or fused, pre-processed, analyzed, and used as training data for a supervised machine learning predictive approach. The data reduction methods are based upon principal component analysis and self-organizing maps to reduce the data feature vector dimensions and to improve supervised classification models. The supervised classification models are based upon artificial neural network, decision tree, and $k$-nearest neighbor methods. The experiment results in high accuracy and kappa rates using this approach. To the best of our knowledge, this is the first technique that relies upon user recognition gestures to predict multiple demographic groups.

## TABLE OF CONTENTS

APPENDIX

# LIST OF FIGURES

Figure Page

# LIST OF TABLES

# LIST OF ABBREVIATIONS

**SMURC** Southern Methodist University Research Compliance

**User-ID** User Identification Number

**MANOVA** Multivariate Analysis Of Variance

**API** Application Programming Interface

**FAR** False Acceptance Rate

**FRR** False Rejection Rate

**EER** Equal Error Rate

**IRB** Institutional Review Board

**ANN** Artificial Neural Network

$K$**NN** $K$-Nearest Neighbors

**PCA** Principle Component Analysis

**SOM** Self-Organizing Map

**CT** Curve-Touch

**UI** User-Interface

**SD** Secured-Digital

**DT** Decision Tree

**ML** Machine Learning

**MF** Multi-Feature

# ACKNOWLEDGMENTS

I would like to thank my family and my friends for their support and suggestions throughout the research and writing of this work. I am also grateful for the guidance and support of my advisor, Dr. Mitch Thornton, who have given me the opportunity of working in the interesting area of the biometric security of smart devices. Special thanks are due Dr. Ted Manikas for his time, advice, and support of this research. Additionally, I am glad to have the support of a famous member of the Department of Electrical Engineering, Dr. Delores Etter, who provided valuable discussion and offered the smart idea of developing an application for collecting case study subjects during the research experiment. I am thankful to Dr. Sukumaran Nair and Dr. Frank Coyle for their intellectual encouragement and for agreeing to be on the dissertation supervisory committee. Finally, special thanks to the test subjects for their valuable time in providing data and participating in the study.

# Chapter 1

# INTRODUCTION

Many smart device applications contain capabilities and features that require the use of private and sensitive user demographic information that is subsequently stored directly on the smart device or within the cloud. The presence of this sensitive data represents a target for malicious exploitation and causes increased security concerns among the user base. As an example, numerous smart device applications rely upon location-based services. Location-based application services allow users to explore, search, and share geographic information with stakeholders. Furthermore, industries or other third-parties can also create and serve customized advertisement services based upon this and other demographic data. However, these services typically require smart device users to register accounts that involve entering personal demographics information to enable sharing their experiences with other new users and for developing friend recommendation services (Alharbi and Thornton, 2015, 2016). Due to increasing privacy concerns and generally, the increased awareness among the user base of issues regarding computer security and privacy, applications that require such personal data to be disclosed are often provided with intentional inaccuracies, in a limited fashion (i.e., not allowing the location service to be enabled), or not used at all. The research presented here investigates alternative means for an application to determine user demographic data. Specifically, the extraction of user demographic information from data inherently associated with user gestures of smart devices rather than requiring explicit disclosure is considered. The rich and diverse set of environmental sensors present in modern smart devices serve as generators of data that can be fused into a feature vector and subsequently used in a supervised learning approach to inherently predict the demographic classification of a particular user when explicit demographic information is not provided, or to detect the presence of inaccurate explicitly provided data. This inherently determined user demographic information can also be used to validate or authenticate explicitly disclosed demographic data when present, or to replace such data when it is not present.

Based upon our results, demographic user classes can be distinguished through unique characteristics of the gestures they use when interacting with smart device applications. Each gesture carries or comprises individual behavior recognition patterns that can be distinguishable from those of a different class. By implementing machine learning techniques based on the behavior recognition patterns, we created an automatic and effective demographic group predication mechanism. Our mechanism can predict user demographic information such as gender, age, native language, nationality, etc. that can be used to serve several demographic services and goals without requiring a user to enter explicit demographic information into their device.

## 1.1. The Study Contributions

Motivated by the above, our work has resulted in several contributions that are described in the following subsections.

### 1.1.1. The Considered Demographic Groups

Our contribution considers twelve demographic groups that representative of the one hundred test subjects in our study. Figure 1.1 represents the twelve demographic groups in a set of bar charts. Each chart shows the true distribution of test subject data with bars that have lengths proportional to the value that they represent. The bar lengths represent a percentage of membership within a particular demographic group. The percentages are calculated based on the total number of users. For example, the gender bar chart has two bars which are the male and female users, and their percentages are 68% and 32%, respectively. Moreover, the names of the eleven demographic groups are given in the figure as titles for each chart.

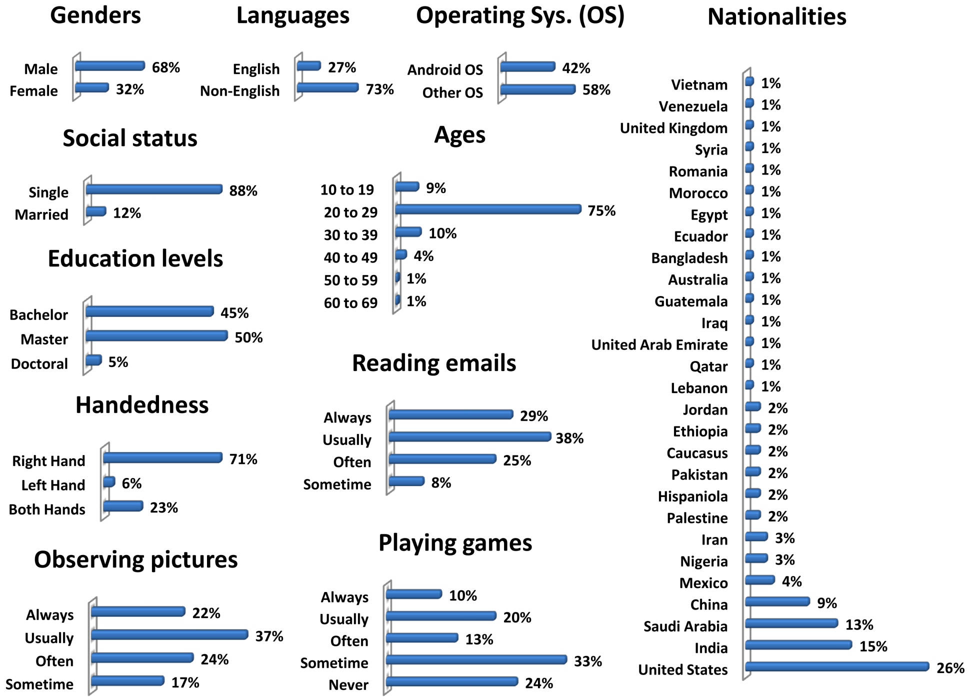


Figure 1.1. Subject Demographic Group Histograms

For conciseness, we encoded every user demographic group as D1 through D11 since we have a multiple number of groups as shown in Table 1.1. This Table also provides the cardinality of each of the demographic group, which is equivalent to the number of bars that are presented in Figure 1.1.

### 1.1.2. Biometric Approach

Biometrics approaches have a similar but different goal as compared to demographic classification. The goal in biometrics is usually the authentication of a particular person whereas our goal is to infer certain demographic classes in which that particular person is a member. Nevertheless, due to the similar nature of the demographic classification and the biometric identification problems, we benefit from the results of the biometric research

Table 1.1. Number/Labels of the Classification Problem

| D# | Demographic | # of the classification problem |
|---|---|---|
| **D1** | **Genders** | 2 |
| **D2** | **Languages** | 2 |
| **D3** | **Operating System** | 2 |
| **D4** | **Nationalities** | 28 |
| **D5** | **Ages** | 6 |
| **D6** | **Social status** | 2 |
| **D7** | **Education levels** | 3 |
| **D8** | **Handedness** | 3 |
| **D9** | **Reading emails** | 4 |
| **D10** | **Observing pictures** | 4 |
| **D11** | **Playing games** | 5 |

community. We can divide biometric approaches into two categorizes: dynamic-based and behavioral-based approaches. Most of the studies in (de Mendizabal-Vzquez et al., 2014; Frank et al., 2013; Kwapisz et al., 2010; Maxion et al., 2010; Sitová et al., 2016) concentrate on using the dynamic-based approach. The dynamic approach calculates extracted features and introduces a new set of features based on generic statistical measures such as mean, median, standard deviation, distance, speed, acceleration, etc.. However, this approach requires more calculation time when used in Machine Learning (ML) techniques. Alternatively, some studies in (Ahmed and Traore, 2007; Lin et al., 2013; Muaaz and Nickel, 2012) use the behavioral-based approach because it is a faster and simpler approach. This approach acquires extracted features and immediately applies them to ML techniques based on the previously collected data as it was defined in the application without extracting any new features. Both approaches are used to recognize users in a short period of time. They depend on the amount of behavioral data that is available to increase performance and accuracy: the more data there is, the greater of the increase in performance and accuracy will be (Frank et al., 2013; Lin et al., 2013). Our work utilized the behavior-based approach because it was hypothesized to be more effective, feasible, and did not require excessive calculations.

### 1.1.3. Multiple Sensors Method

An advantage of our study is that modern smart devices include a large variety of sensors that are already present within the device and do not require the use of new or customized interfaces to obtain other data. Our study contribution is to utilize three common sensors including the touchscreen keystrokes, non-keystroke touch-screen interactions, and the internal accelerometer. We chose these three sensors because they were determined to provide the most useful data and are available in most devices. However, other studies were based on a more limited number of sensors such as in (Allen et al., 2011; Frank et al., 2013; Lin et al., 2013; Muaaz and Nickel, 2012; Thornton, 2011). Another difference in our work as compared to past investigations is that they were focused on user authentication. In contrast, our study uses multiple sensors to build more predictive classification models and focuses on the classification of demographic groups.

### 1.1.4. Machine Learning Methodology

Our contribution is to devise a methodology that mainly depends on machine learning (ML) techniques to predict user demographic information. Figure 1.2 illustrates the study's methodology diagram. Our application has seven user-interfaces (UI)s, where each of the UI generates user gesture data inputs corresponding to keystrokes, other touch-screen interactions, and accelerometer sensor data. The methodology implements a data integration process that integrates all the data features that were obtained from the different UIs into a single data frame. We refer to this single data frame as the multi-feature (MF) data set because the gesture data originated from various sensor inputs. The methodology then implements a data pre-processing process that contains pre-filter functions. Some of the functions are optional (Colored in gray), and the others are primary. After pre-filtering, the methodology applies the data reduction process by using the principal component analysis

(PCA) technique to reduce the user pre-processed data dimensions into a reduced number of new dimensions. Next, after the MF dimensions are reduced, the methodology applies the self-organizing map (SOM) technique to compress the new dimension vectors and organize it into fixed grids. Finally, the supervised learning data set splits into training and testing data sets. The training data set is used to train the supervised classification process, which depends on three classifiers to compare between them in performing demographic predictions. The classifiers are artificial neural network (ANN), decision tree (DT), and $k$-nearest neighbors ($K$NN) classifiers. The testing data set is used to evaluate the trained models and to produce accuracy results. Note that all of the methodology processes (integration, pre-processing, and reduction) are performed separately for each of the user data. Chapter 6 provides and explains the sequence of the methodology process operations in detail.

### 1.2. Project Method and Goals

In this section, we provide a high-level overview of our general idea and our goals for the demographic group prediction mechanism as shown in Figure 1.3.

Users inherently interact with the built-in embedded sensors in their smart devices while they accomplish various activities. The sensors capture data during the interaction, which is monitored and collected in the smart device's local database storage. When the data is collected in the smart device database, it can be directly forwarded to pattern recognition algorithms in order to detect and confirm whether the operation is executed by the appropriate user or not. If the pattern is confirmed, the resident behavioral pattern recognition component computes the difference between the currently collected data and the stored patterns for future analysis and computational purposes. If a pattern is not confirmed, it could mean that a new user pattern has been discovered. The system will update the new pattern and categorize it as belonging to different demographic group or groups. All the stored patterns resulting from the training phase that are present in the smart device database can be retrained to achieve faster access for the dependent applications. However, if there are a large

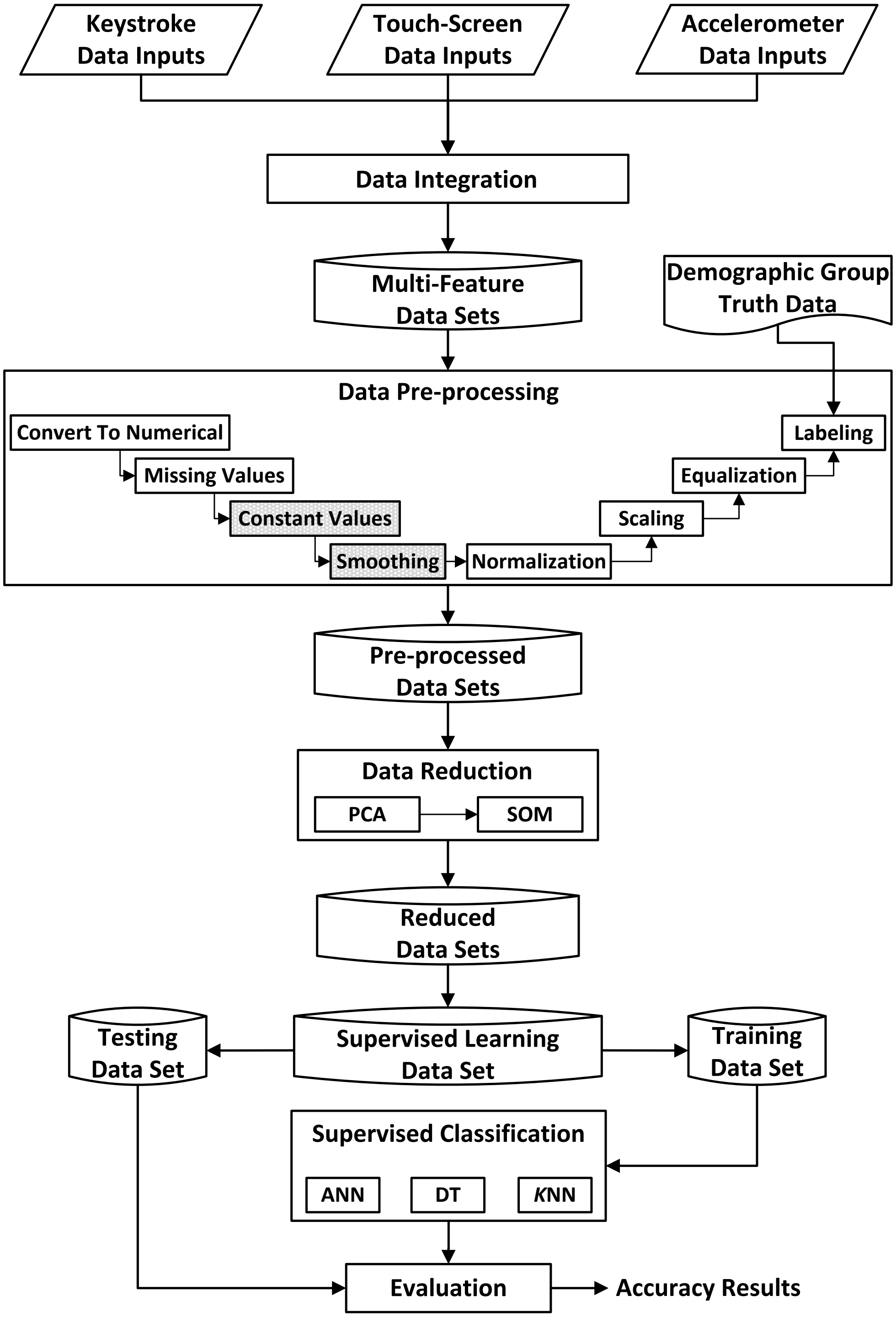


Figure 1.2. The study's Methodology Diagram

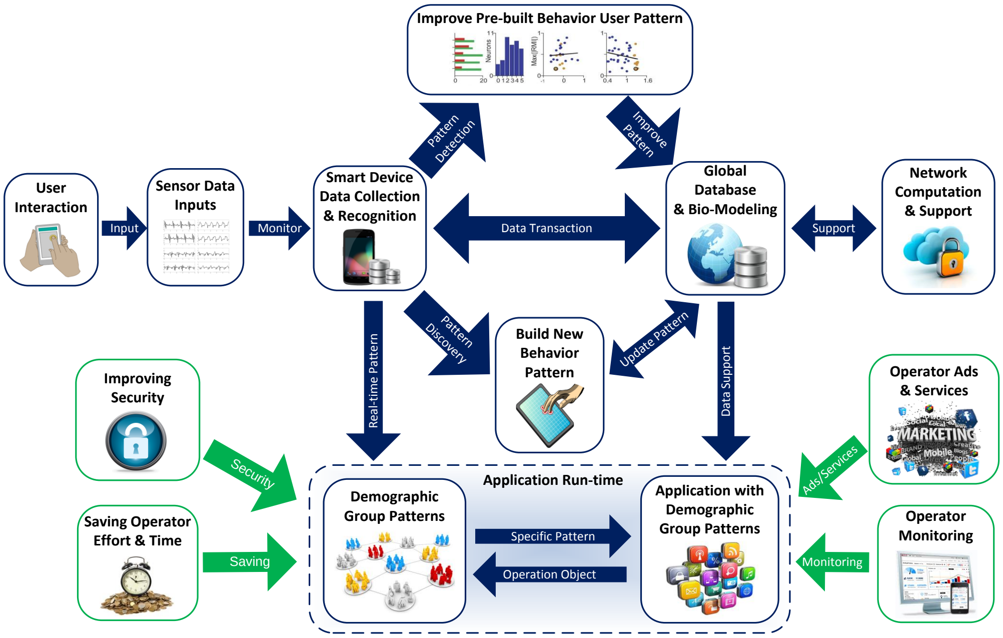


Figure 1.3. Demographic Group Prediction Mechanism General Idea and Goals

number of patterns, only the most frequently used patterns can be stored in the smart device storage system, while other patterns can be kept in the operators' cloud storage. When the application is enabled to have a real-time internet connection, it can update the patterns automatically by using application background hiding techniques without interrupting the user's usage of a particular application (Nixon et al., 2013a).

Using the demographic group prediction mechanism opens a pathway to several goals. First, mobile device or service operators can send suitable categorized advertisements and services corresponding to the appropriate demographic group. Second, some operators or authoritative entities can monitor users in order to identify, classify, or understand other characteristics of a user without prior training data obtained specifically from that user. Thirdly, when a user is entering explicit demographic information, it can be validated by comparing it with the inherently computed demographic information provided by the predictive classifier. Another goal is security. If the device operators or some other supervising authority notice any behavioral changes indicating that an unauthorized user may be masquerading as an authorized user, they can take appropriate action. Operators or device owners can also use the proposed mechanism as an authentication technique in a dynamic and continuous manner rather than the more common static technique provided by log-in procedures. Finally, many operators exert effort and spend time collecting customer and employee demographic information in order to make their business investments more suitable for certain demographic groups. This method can be used instead of or to augment this effort and to decrease the time requirements of this effort to predict demographic information from the use of customer and employee smart device applications.

## 1.3. The Study Structure

This thesis is structured as follows: Chapter 2 explains some of background concepts that are necessary for the work. Chapter 3 surveys briefly the most significant published related work. Chapter 4 focuses on the application that is used in the data collection process. Chapter 5 describes the data acquisition process and design considerations. Chapter 6 discusses the proposed methodology. Chapter 7 analysis the collected data. Chapter 8 presents the experimental results. Finally, Chapter 9 discusses the work limitations, potential extensions, final thoughts, and future work.

# Chapter 2

# BACKGROUND

This chapter provides background for fundamental concepts on biometric , machine learning, and smart device applications.

## 2.1. Biometrics

Biometrics involves the identification and verification of the personal characteristics of the individual who is using the device. There are two biometric methods that are typically divided into two denominations: behavioral and physiological biometrics. Behavioral biometrics is based on the way people interact with a device while performing certain functions such as keystroke dynamics and mouse movement. Physiological biometrics is based on bodily characteristics such as facial recognition, fingerprints, and iris scanning (Bhattacharyya et al., 2009). Furthermore, behavioral biometrics is useful when attempting to implement a form of continuous user authentication, which makes the authentication system non-intrusive and ensures an active authentication. Our work concentrates on the behavioral type.

All biometrics are measured under certain criteria, known as biometric performance measures. Many researchers have plotted and studied biometric performance and evaluation measurements. One well-known performance measurement is the false acceptance rate (FAR), which measures how often a meddler is likely to successfully bypass the biometric verification or authentication transaction. A lower rate is more secure; for example, a FAR of 1% indicates that the chance of fooling the system is 1:100. Closely related to FAR is the equal error rate (EER), which describes the relationship between FAR and FRR which is converse, although not always linearly in behavioral biometrics. EER represents where the FAR and

FRR would be equal. The best technologies have the lowest ERR rate (Bhattacharyya et al., 2009; Vacca, 2007). Another well-known performance measurement is the false rejection rate (FRR). FRR signifies how often a real user is unable to be verified successfully. A high rate translates into more user retries; hence, usability suffers. The final testing technique is the receiver operatic characteristic curve. The curve is a characteristic graph of the biometric system where the x-axis correspond to the threshold of the biometric system, which is the rate of false positives (accepted impostor attempts). On the other hand, the y-axis corresponds to both the FAR and FRR values, which are true positives (genuine attempts accepted) (Bhattacharyya et al., 2009).

## 2.2. Machine Learning

ML is a broad field of artificial intelligence that aims at mimicking the intelligence abilities of humans with machines. In the last few years, much of ML research has been greatly expanded and has produced a large number of very accurate and efficient algorithms. ML has been used to better understand how to make machines able to "learn" and develop a mechanism for adapting to unknown situations by splitting the collected data into training and testing data sets. There are two types of ML : unsupervised and supervised learning. Unsupervised learning seeks to uncover hidden regularities or detect anomalies in the data (e.g. clustering). Supervised learning discovers solutions for classification and regression problems. Classification problems exist when a particular data set has a classed label for each associated instance. Otherwise, regression problems occur when the data set has real valued labels (Rätsch, 2004). ML has many primordial algorithms which are in regression algorithms (e.g. regression trees and ridge regression) and unsupervised learning algorithms such as clustering and principal component analysis. Additional algorithms are in online learning, reinforcement learning, model-selection, and feature-selection. These algorithms has spread the applicability of ML algorithms drastically in much research (Rätsch, 2004).

Classification as a principle may indicate categorization, an approach in which ideas and things are recognized, differentiated, and understood in terms of classes or groups. The definition of classification is ”a data mining (machine learning) technique used to predict group membership for data instances” (Patel et al., 2013). Classification algorithms are aimed at classifying objects depending upon the nature of the objects themselves. One of the fundamental concepts in machine learning is classification, also referred to as pattern recognition. There are two techniques of classification. One technique is model construction, which predicts categorical class labels as either discrete or nominal. The second technique is model usage, which classifies data or constructs a particular model based on the training set and the values or class labels of a classifying attribute and uses the classified values to classify new data. On the other hand, there are many classification algorithms that have been studied such as k-nearest neighbor, linear discriminant analysis, decision tree, neural networks, support vector machines, boosting, and other machine learning algorithms (Rätsch, 2004).

There are many data mining and machine learning tools such as Weka, Orange data mining, and R Studio. Weka is an open scours program written with the Java programming language from the University of Waikato, in New Zealand. Weka is a very powerful data mining tool which has a collection of machine learning algorithms (Hall et al., 2009). Orange data mining is another ML tool similar to Weka. It is, however, stronger than Weka in data visualization. Orange is also an open scours program that is written in C++ and python programming languages. Orange has a lot of plots and data visualization such as histogram, scatter, and polyviz plots (Demšar et al., 2004). Another important data mining tool is R Studio. R Studio is a statistical build-in model as well as a data mining model. R studio use a command line to operate these models (Williams, 2009). All of these tools include data mining principles such as data pre-processing, filtering, classification, principal component analysis, regression, clustering, association rules, visualization and, more importantly, it can create new machine learning schemes.

## 2.3. Smart Device Application

The work depends on gathering the user biometric behavioral data gestures while using smart device application. The study developed a simple Android application using the Android SDK Developer Tool. Android is a software stack for Android devices that includes an operating system, middle-ware, and key applications. The Android SDK provides the programming tool and APIs necessary to begin developing applications on the Android platform using the Java programming language. Java is one of the object oriented programming languages that allows programmers to create objects and allows them work together. Objects are treated as an entity which drives the class functions and attributes to where they belong. An object oriented model is a collection of interactions of various types of objects, which is different from conventional programming. Moreover, instantiation is the process of creating an object. Java has all the object oriented concepts and features such as polymorphism, data encapsulation, and inheritance and more. There are three characteristics of Java that are used in this study data collection application which are simplicity, robustness, and multi-threading (Arnold et al., 1996). Therefore, Android was built from the ground-up to enable developers to create compelling mobile applications that take full advantage of all a handset has to offer. It was built to be truly open and countless studies have utilized this tool (Meier, 2012). In order to collect the user data gestures while using the application, the study utilized SQLite database. SQLite is an in-process library that implements a self-contained, serverless, zero configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is currently found in more applications, including several high-profile projects (Owens and Allen, 2010).

Chapter 3

# OTHER RELATED WORK

The chapter discuses the relationship between demographic and authentication classification. Additionally, it briefly surveys some of the latest related work of smart device's authentication. Finally, the chapter concludes and provides further reading for the most significant publication papers.

## 3.1. Relationship Between Demographic and Authentication

Other work uses data similar to that we use in our work, but for the purpose of human authentication as opposed to demographic classification. Authentication is related to the problem of demographic classification but with several important differences. The relationship between demographic classification and authentication is that in the latter case, the goal is to identify a user as being a member of a set with unity cardinality. That is, the authentication problem attempts to classify a user as being a member of a set containing exactly one element and whereby the collection of all possible sets is disjoint (i.e., all possible set intersections result in the null set). In contrast, the demographic classification problem classifies users as belonging to a set generally composed of many members and whereby the collection of all possible sets may be non-disjoint. For example, both males and females may also be members in the set of native English speakers. From this point of view, the universe of authentication sets is one where each individual is represented by a set containing one element that uniquely identifies that individual whereas the universe of demographic classification sets is comprised of sets that have multiple elements and whereby an individual may have membership in more than one set.

Figure 3.1 illustrates the relationship between demographic and authentication classification concepts. The left side of the Figure shows a set of data point in the space, where the smart device learner considers to classify the data points based on a few number of demographic groups that were randomly color coded. By classifying more number of the demographic groups for the same the data space, some of the data points would overlap to discover other demographic classification ranges. For example, some data points can be used to classify the user gender and in the same time can be use to classify the user age. Furthermore, if the smart device learner be able to classify many number of the demographic groups, it would reach to the point that might authenticate all the data point that belong to a specific user.

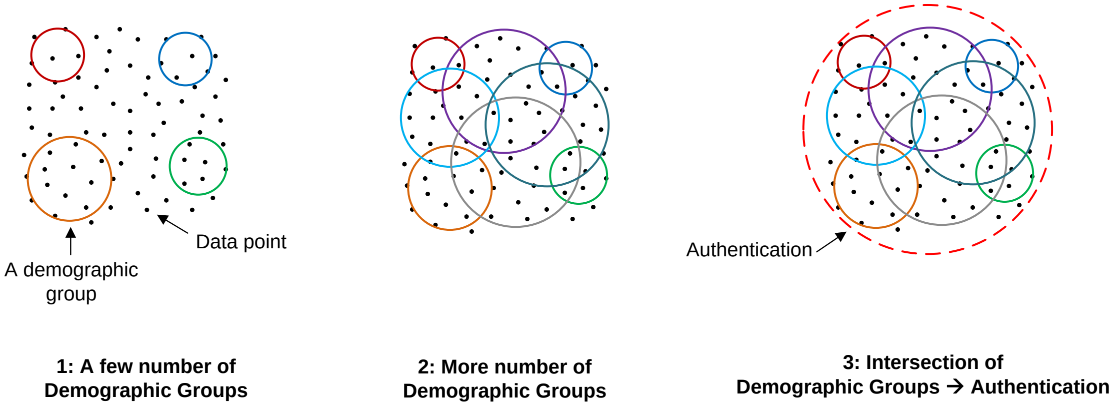

Data point
A demographic group
Authentication
1: A few number of Demographic Groups
2: More number of Demographic Groups
3: Intersection of Demographic Groups → Authentication

Figure 3.1. The Relationship Between Demographic and Authentication Classification

### 3.2. Surveyed Work

The concept of user demographic group predication mechanisms is discussed in some past studies that focus on classifying user's demographic from their writing and speaking styles as opposed to their use of mobile smart devices. The studies by Berryman-Fink (Berryman-Fink and Wilcox, 1983) and Simkins-Bullock (Simkins-Bullock and Wildman,

1991) classified the user genders from the user writing styles. The Biber work (Biber et al., 1998) has found a significant difference between user gender in language structure based on a correspondence corpus. Recently, there have been a number of studies that concluded that users with similar demographic group behavior might visit similar web-pages. The study by Hu et al. (Hu et al., 2007) proposed an approach to predict user demographics such as gender and age from the user web browsing behaviors.

The work by Ying et al. (Ying et al., 2012) is the only study we found that could be more closely compared to our work. They proposed a demographic group prediction approach based on user mobile behaviors. They used the mobile data challenge (MDC) data set that was collected from a Nokia device. They developed a dynamic framework to compute a total of 45 features. Later, they applied a multi-level-classification models technique on the computed features. This technique was used to build several models. The models were validated to find the best accuracy result among their proposed models. Then, it used a feature selection method to find the best feature set for every model. In the end, the method integrates all of the models to perform demographic predictions. For example, with regard to marital status, they selected a logistic classification for level 1 and $K$NN for level 2, where they achieved accuracy rates of 71.30% and 79.67%, respectively. However, their data set was formulated as two or three distinct classification problems. They also did not include user gesture data obtained from the touchscreen nor did they use the accelerometer. Finally, they considered only five demographic groups to study. In contrast, our work considered more demographic groups and proposed only one classification model to predict all demographic groups.

There are a large number of studies regarding the concept of user authentication mechanism. The remainder of this section briefly discusses how other past studies applied ML techniques similar to those in our work. This section categorized the surveyed work as keystroke, touch-screen, and accelerometer work.

### 3.2.1. Keystroke Work

Thornton (Thornton, 2011) characterized keyboard dynamics as the characteristics of writing or cadence patterns of arrangements of keystrokes on a keyboard. Thornton also explained that the musical writing patterns of diverse individuals had characteristics that are one of a kind. Due to the fact that writing requires an interaction of the psyche and manual smoothness, many complicated physiological methodologies were included in the chain of occasions starting with having a contemplated which particular key on a keyboard is to be discouraged (and released) and the actual occasion of key passage.

One of the significant work that addressed the concept of the user keystroke authentication was reported by Allen et al. (Allen et al., 2011). The management of password is a big issue for the user as well as supplier. They invented a device for the classification of the user by recognizing the pattern of the input. They implemented a method based on two features of user keyboard gesture; dynamic timing fused with key pressure. They used a special keyboard that included embedded pressure sensors and combined the pressure data with traditional dynamic keyboard timing data. The device was composed of an input device having a user input interface, sensors to sense the pattern and a memory device to store the data. The device was used to measure the pressure and time of input of the original users and store it. Whenever another person inputs the data, the pressure and time was compared with the values of the original user. This invention provided the solution by identifying the pattern in which a genuine user inputs own password.

Their work resulted in an authentication approach that required the use of a keyboard capable of measuring keypress pressure. They achieved to 10% of the FAR and 0% FRR. The results can be improved by using the EER or ROC curve methods. These methods provide a reliable method, but main concern is to reduce hamming distance and analyzing only those values with small hamming distance. They reported that the pattern recognition system can be used for encryption for different scenarios and classifying them on the basis of sex, gender, age and dominate typing hand. This method can be used to a non-transitory readable medium having one or more segments that can perform the sets when executed by one or more processors. The major prospects of the invention have been discussed in the report but still some more comprehensive changes can be done to make this method more reliable and more consistent.

Other researchers focused on smart devices as tablets and cell-phones. Maxion et al. (Maxion et al., 2010) work was based on the data collection from twenty eight participants who typed the same ten-digit numbers using only the right-hand index finger on an IBM ThinkPad X60s notebook computer. They proposed an approach that used statistical machine learning classification algorithms, in particular the Random Forest classifier. They achieved with excellent an un-weighted FAR of 99.97% and FRR of 1.51%, using practiced 2-of-3 encore typing with outlier handling. Finally they suggested to use this level of accuracy approach as a two-factor authentication for passwords or PIN numbers.

Joshi et al. (Joshi and Phoha, 2007) also discussed this matter. They proposed a neural architecture model to authenticate computer users through keystroke dynamics. The architecture consists of a set of SOM maps where each user has a distinguished map. Each map has an array of neurons in the input layer. They achieved a 97.83% authentication accuracy result. The study discussed by Dozono et al. (Dozono and Nakakuni, 2008) also applied the SOM method to integrate multi-modal behavioral vectors that were collected from keystroke timings and handwritten patterns using both the computer keyboard and a touch panel.

Similarly, the work by Sinthupinyo et al. (Sinthupinyo et al., 2009) implemented the SOM method to construct one map for each user pattern. They classified the maps using an ANN (back-propagation) and the DT (J48) classification algorithms. They compared both classification methods using a 10-fold cross-validation strategy. They achieved accuracy rates of 48.05% by implementing the DT classifier and 54.10% by implementing the ANN classifier. Furthermore, Mendizabal-Vzquez et al. work (de Mendizabal-Vzquez et al., 2014) provided a method to improve the ANN classifier performance by applying the PCA data reduction algorithm.

### 3.2.2. Touch-Screen Work

Researchers have also studied the touch-screen data gestures. One significant study by Frank et al. (Frank et al., 2013) investigated whether the studied classifier was able to consistently authenticate users based on the way they interacted with the touch-screen of a smart phone. They proposed management of thirty behavioral touch features as a framework and then used it as a behavioral pattern. In a systematic examination intended to test how this behavioral pattern displays consistency over the long run. They gathered touch data from users interacting with a smart phone utilizing basic navigation maneuvers, i.e., updown and leftright scrolling. They collected data from forty-one users. They used weighted $K$NN (W$K$NN) and support vector machine (SVM) classifiers that learned the touch behavior of a user amid an enlistment phase and has the capacity accept or reject the current user by monitoring interaction with the touch-screen. They achieved a median EER of 0% for inter-session authentication, 2% to 3% for inter-session authentication, and beneath 4% when the authentication test was carried out one week after the enlistment by using the W$K$NN classifier.

Lin et al. (Lin et al., 2013) also discussed touch-screen data gestures. They pointed out that applications raised new security issues to smart-phone users. On the other hand, the current protection mechanisms of smart-phones were not sufficient because of the convenience issue and shoulder-surfing issue. Therefore, they proposed a non-intrusive authentication approach based on touch-screen of smart-phones. Their work was the first openly reported study that adopts the histogram features of touch-screen to manufacture an authentication model for smart-phone users. They strategy used to categorize the touch screen behavioral data features as histogram of touch position, histogram of touch pressure, and histogram of touch size. The approach also used the W$K$NN classifier where the K nearest training examples around a query sample are determined by KL-Divergence algorithm. The experimental results were validated base on the data sets of the up-down flicks and the left-right flicks in the study's application interfaces. They experimental results had an EER less than

5.5% while the number of touches exceeds 30 and ERR of 2.9% to 3.6% when the number of touches was sixty.

### 3.2.3. Accelerometer Work

Some studies have focused on the behavioral factors of the holding posture of human hands in smart device applications during normal usage. Nixon et al. (Nixon et al., 2013b) proposed a novel mobile user authorization and classification approach based on the recognition user's gesture. They studied the produced data of three volunteers from the three-axis accelerometer and Gyroscope of the mobile built-in sensors. The work main objective for pattern of the user's gestures was, usually recognized and stored in the mobile storage. Every time the person tries to access the mobile phone data, user's gesture pattern was compared with the data of original user and the access was granted or denied. The sensors of the mobile phone usually record the data over time. The sensors could also detect the holding position of the device. They compared to other security solutions such as password, track pattern and finger print. Their analysis demonstrated that gesture could offer a solution for device data protection. However, they approach did not use any classification techniques for the recognition patterns.

Another important related study investigated by Muaaz et al. (Muaaz and Nickel, 2012) wearable sensors, and they used a commercially available mobile device. They identify their user based upon the unique style of user walking. The walking style of a person, termed as gait, depended upon the natural style and surfaces on which he or she was walking. The problem faced in application of gait recognition was that the gait was effected by type of foot wear and surface. They resolved approach to solve this problem based upon the comparison of the two walks of a person. The collected data from forty-eight users where each walk was stored in separate file. The recorded raw data indicated a periodic repetition after several steps, which was called a cycle in order to create templates for each subject, these cycles were extracted. They used filters and data prepossessing such as interpolation, weighted moving average, centering around zero, cycle length estimation, cycle detection, cycle normalization, deletion of unusual cycles, and computation of typical cycles in order to have a unique recognition patterns. They initialed validation using the weighted K-Nearest Neighbor classifier and applied majority voting yielded non-acceptable False Non-Match Rate of over 80% (normal walk) while the False Match Rate was 0%.

Moreover, Sitova et al. (Sitová et al., 2016) proposed a new behavioral biometric features for continuous authentication using the accelerometer and gyroscope of the mobile built-in sensors. They introduced hand movement, orientation, and grasp (HMOG) to generate a set of dynamic features from how a user grasps, holds, and taps on the device. They collected data from one hundred of users under two conditions: sitting and walking. They achieved EER of 7.16% (walking) and 10.05% (sitting). Their analysis demonstrated that gestures could offer a solution for device data protection.

### 3.3. Conclusion

The surveyed work applied various approaches and methodologies for the extracted user data gestures from using the smart device applications (Grenga, 2014). Most of the surveyed work used the ML concepts in order to classify the collected data recognition pattern and measure their proposed approaches using the biometric performance measurements. Although those work being studied in the biometrics and machine learning, they were limited with the number of the sensors to one or two. Another limitation with the surveyed work were only concentrate on the concept of authentication. Some of the studies as in Allen et al. (Allen et al., 2011) mentioned the possibility of the demographic groups classification.

Through the surveyed work, we found that some studies implemented the PCA as a data reduction methods for the extracted features, where other work implemented the SOM as data vector compression method to improve the classification process and results. The most classifiers being used with those two methods are ANN, DT, and $K$NN. Recently, the ML investigators studied on how to improve the SOM method with other methods. The study by Silva et al. (Silva and Del-Moral-Hernandez, 2011) proposed a technique to improve the SOM method by combined it with the $K$NN classifier that it used for classification task. They reported that this technique is much faster than traditional methods, where they achieved classification rates of 91.03%. The study by Wu et al. (Wu and Li, 2014) proposed another technique to further improve the SOM method. The technique was to implement the SOM method based on the PCA dimensionality reduction method and use a weighted Euclidean metric for the $K$NN classifier. This technique is useful for finding the proportion between class variances and within a class variance for the training data samples. It improved the accuracy and classification time results by implementing the PCA method to reduce dimensionality. However, both of the studies applied their techniques on the digits of car plate recognition patterns or real-world data sets, and not on either user authentication or demographic grouping classification.

Below is a supplementary literature list which is a recall of most publication papers are significant for this investigation.

## Supplementary Literature List

Jeffrey David Allen, John Joseph Howard, and Mitchell Aaron Thornton. Method for subject classification using a pattern recognition input device, October 22 2011. US Patent App. 13/279,279.

Cynthia L Berryman-Fink and James R Wilcox. A multivariate investigation of perceptual attributions concerning gender appropriateness in language. *Sex Roles*, 9(6):663–681, 1983.

Douglas Biber, Susan Conrad, and Randi Reppen. *Corpus linguistics: Investigating language structure and use.* Cambridge University Press, 1998.

I. de Mendizabal-Vzquez, D. de Santos-Sierra, J. Guerra-Casanova, and C. Snchez-vila. Supervised classification methods applied to keystroke dynamics through mobile devices. In *Security Technology (ICCST), 2014 International Carnahan Conference on*, pages 1–6, Oct 2014. doi: 10.1109/CCST.2014.6987033.

Hiroshi Dozono and Masanori Nakakuni. An integration method of multi-modal biometrics using supervised pareto learning self organizing maps. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 602–606. IEEE, 2008.

Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *Information Forensics and Security, IEEE Transactions on*, 8(1):136–148, 2013.

Anthony J Grenga. Android based behavioral biometric authentication via multi-modal fusion. Technical report, DTIC Document, 2014.

Jian Hu, Hua-Jun Zeng, Hua Li, Cheng Niu, and Zheng Chen. Demographic prediction based on user's browsing behavior. In *Proceedings of the 16th international conference on World Wide Web*, pages 151–160. ACM, 2007.

Shrijit S Joshi and Vir V Phoha. Competition between som clusters to model user authentication system in computer networks. In *2007 2nd International Conference on Communication Systems Software and Middleware*, pages 1–8. IEEE, 2007.

Chien-Cheng Lin, Chin-Chun Chang, and Deron Liang. A novel non-intrusive user authentication method based on touchscreen of smartphones. In *Biometrics and Security Technologies (ISBAST), 2013 International Symposium on*, pages 212–216. IEEE, 2013.

Roy Maxion, Kevin S Killourhy, et al. Keystroke biometrics with number-pad input. In *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*, pages 201–210. IEEE, 2010.

Muhammad Muaaz and Claudia Nickel. Influence of different walking speeds and surfaces on accelerometer-based biometric gait recognition. In *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on*, pages 508–512. IEEE, 2012.

Kent W Nixon, Xiang Chen, Zhi-Hong Mao, Yiran Chen, and Kang Li. Mobile user classification and authorization based on gesture usage recognition. In *Design Automation Conference (ASP-DAC), 2013 18th Asia and South Pacific*, pages 384–389. IEEE, 2013.

Leandro A Silva and Emilio Del-Moral-Hernandez. A som combined with knn for classification task. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2368–2373. IEEE, 2011.

Jennifer A Simkins-Bullock and Beth G Wildman. An investigation into the relationships between gender and language. *Sex Roles*, 24(3-4):149–160, 1991.

Sukree Sinthupinyo, Warut Roadrungwasinkul, and Charoon Chantan. User recognition via keystroke latencies using som and backpropagation neural network. In *ICCAS-SICE, 2009*, pages 3160–3165. IEEE, 2009.

Zdeňka Sitová, Jaroslav Šeděnka, Qing Yang, Ge Peng, Gang Zhou, Paolo Gasti, and Kiran S Balagani. Hmog: New behavioral biometric features for continuous authentication of smartphone users. *IEEE Transactions on Information Forensics and Security*, 11(5):877–892, 2016.

Mitchell A Thornton. Keyboard dynamics. In *Encyclopedia of Cryptography and Security*, pages 688–691. Springer, 2011.

Jiunn-Lin Wu and I-Jing Li. The improved som-based dimensionality reducton method for knn classifier using weighted euclidean metric. *International Journal of Computer, Consumer and Control (IJ3C)*, 3(1), 2014.

Josh Jia-Ching Ying, Yao-Jen Chang, Chi-Min Huang, and Vincent S Tseng. Demographic prediction based on users mobile behaviors. *Mobile Data Challenge*, 2012.

# Chapter 4

# DEMOGRAPHIC GROUP APPLICATION

Every day smart device companies add newer sensor or remove excited sensors from smart devices to improve the smart device user usage features. Modern smart devices are equipped with a large variety of environmental sensors that can capture the user behavioral patterns. The number of sensors depends on the smart device types and versions and the hardware capability supports (Lane et al., 2010). Figure 4.1 presents the majority of smart device excited sensors.

Many of the related studies have leverage for choosing some of the standard components of smart device sensors. Our selection mechanize of the smart device sensors considered choosing the most commonly used sensors, which are keyboard, touch-screen, and accelerometer sensors. There are several reasons behind our selection, which are as follows:

- Security: The camera, microphone, and GPS sensors capture significant users features could be useful information for the attacker in case the system was compromised. In this case, the attacker would have the user location, face, and voice features.

- Efficiently: There some sensors depend on other technologies which may affect the efficiency of the approach. For example, the GPS sensor depends on the internet coverage having access to the user's locations.

- Natural Environmental Factors: The camera sensor is easily affected by natural environmental factors like the light and brightness associated with the user's pictures. Similarly, the microphone sensor is affected by other environmental sounds.

- Rapid Industrial Development: Smart device companies introduce new features to some sensors which might impact the process of extracting data.

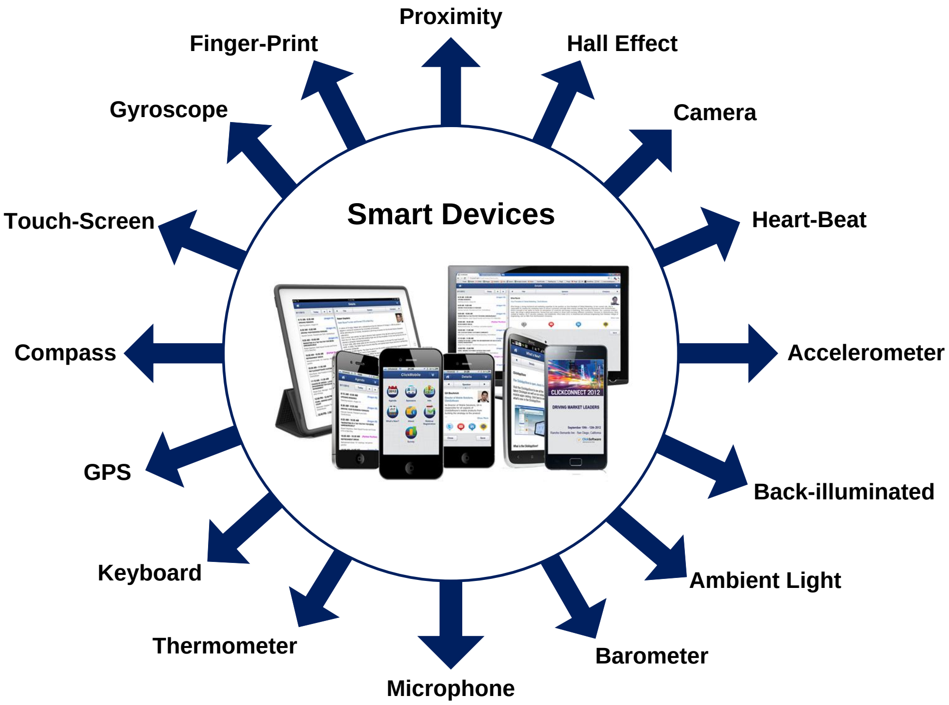


Figure 4.1. The Smart Device Excited Sensors

- New Sensor Technologies: Some older smart devices hold outdated technology sensor features that means approaches based on new devices may not be applicable.
- Smart Device User Usages: Users use some sensors more frequently than others. For example, touch-screen sensors are the most frequently used sensor because most smart device applications are touch based.
- User Privacy: Some sensors capture private user feature data such as camera, microphone, and GPS. Smart device users would like to have their data secure and not be shared with any other system.

This chapter focuses on the developed application that was used in the data acquisition process. The chapter provides the application UIs followed by discussion of the security and hiding methods and database management features. Finally, it discusses the extraction behavior features.

### 4.1. The application UIs

Our pilot study was carried out to develop an Android-based demographic data collection application that included seven different user exercises. This application, named “Classy,” was deigned to be User Interface (UI) friendly and simple as well as an enjoyable game-like experience. In order to make the application friendlier and more interactive for participants. The application requested the human subject to engage in tasks such as entering routine information, re-typing sentences, re-typing random character patterns, reading an article and answering questions about it, drawing on the touchscreen with the user’s finger, zooming in/out on a picture using two fingers, and playing a game that involved spatially reorienting the device. Table 4.1 describes each of the application tasks implemented beneath the Classy UI.

### 4.2. Extracting Features

All Android applications have the capability to run application programming interface (API) packages (Meier, 2012). Part of these APIs can be used to extract user gesture data. Our application used the Text Watcher API to capture the key that the user presses on the soft keyboard. The application also used the Motion Event and On Touch Listener APIs to capture the test subject’s finger movements over the touch-screen device. The Sensor Event, Sensor Event Listener, and Sensor Manager APIs were used to obtain the user gesture movements while holding the captured device tool. Moreover, the application was designed with asynchronous task techniques to increase the performance of the API to run threads in the background of the application because we did not want the user to notice any delay while using the application. The technique by which the application extracted user gesture features is described in more detail in the following section.

Table 4.1. The Demographic Group Application UI Description

| UI# | UI Name | Description |
|---|---|---|
| **1** | **User Information** | This screen asks the user to enter normal information such as name, phone number, email address, city, and zip code. |
| **2** | **Quote** | This screen asks the user to re-write one fixed quotation. |
| **3** | **Tokens** | This screen asks the user to retype three fixed complex strings which include letters, numbers, and punctuations. |
| **4** | **SMU Article** | This screen asks the user to read an article and swipe to the right in order to answer five related questions. |
| **5** | **Hidden Secret Game** | This screen asks the user to answer three questions related to a given picture. The user must zoom in/out to find the answers to these questions. |
| **6** | **Drawing** | This screen asks the user to draw anything the user would like such as a house, any word, or any signature with their finger on the touchscreen. |
| **7** | **Avoider Game** | This screen asks the user to pay a reoriented game. The idea of the game is to cause the Avoider character to move away from displayed black clouds and to eat yellow suns in order to win this game. The Avoider character is moved by reorienting the position of the smart device relative to the horizontal plane. |

### 4.2.1. Keystroke Sensor

Smart device users normally use the application's soft keyboard while they are typing to text, chat, or email someone. All users have their own way of striking the keys. Observing each user's key-striking behavior, allows each user's unique behavioral interaction with the keyboard to be captured. Every key-press associated with timing is different than another key-press time, and thus a timing signature parameter can be generated (Maxion et al., 2010; Thornton, 2011). Figure 4.2 shows an example of a user who typed the sentence "smu @ taxes" where every key had a different press-time during the typing process.

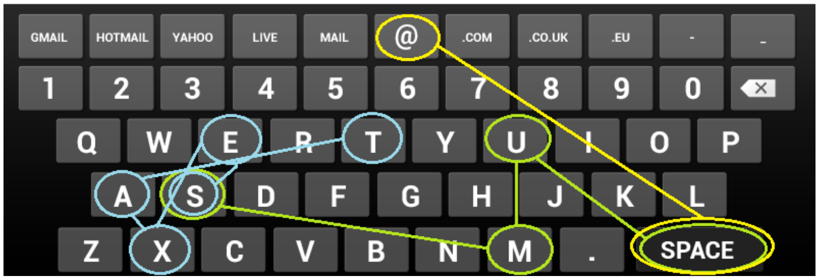


Figure 4.2. Soft Keyboard Key Typing Example

The application captured the keystroke features from UI# 1, 2, and 3. The keystroke sensor has a total of three features. The keystroke actions (F1) contains three keystroke actions referred to as 'before,' 'on,' and 'after' while a particular letter is being typed. Each of the keystroke actions is associated with a keystroke code (F2) that is the ASCII value of the depressed letter. Each of the keystroke actions has time-stamp (F3) in milliseconds (Maxion et al., 2010) that is also used.

Examples of keystroke features data are in Table 4.2. They captured the data entries during the recording process on the application. The example table shows the type of keystroke action taken over the keyboard and indicates whether that action is 'before,' 'on,' and 'after.' If the user, for example, strikes the letter "S" the application stores the letter "S" as On because it is the current letter being pressed. The key the user struck immediately before ”S” is stored as Before, while the letter struck immediately after ”S” is stored as After. The table also shows each action has its own key letter, key code, and key time.

Table 4.2. Keystroke Feature Data Structure

| **KS_Actions** | **Key_Letter** | **Key_Code** | **KS_T** |
|---|---|---|---|
| On | W | 119 | 34288405 |
| After | W | 119 | 34288503 |
| Before | W | 119 | 34288611 |
| On | E | 101 | 34288660 |
| After | E | 101 | 34288707 |
| Before | E | 101 | 34288746 |
| On | S | 115 | 34288796 |
| After | S | 115 | 34288837 |
| ... | ... | ... | ... |

### 4.2.2. Single-touch-screen Sensor

Everyone has a unique way of touching a smart device screen, and the behavioral observations of the touches might be categorized as specific types: Flick, Spread and Pinch, and Drag touches. The flick touch gesture is used to move the scroll-bars of smart device applications such as in email and on browsers. The spread and pinch touch gestures are used separately to zoom-in and zoom-out on the touch screen. Based on previous research observations, flicks and spread and pinch are frequently and commonly used touch gestures while operating smart device applications (Lin et al., 2013). Figure 4.3 shows a perfect example of eight users' touch interaction recorded data with a smart device touch-screen (Frank et al., 2013).

The research discovered that the user always generates touch actions while touching a screen with a finger. This occurs when a user touches the screen and moves from position A to position B (i.e. from right to left, left to right, up to down, or down to up). The touch actions creates what the study called a Curve Touch (CT). A CT gets created from the time

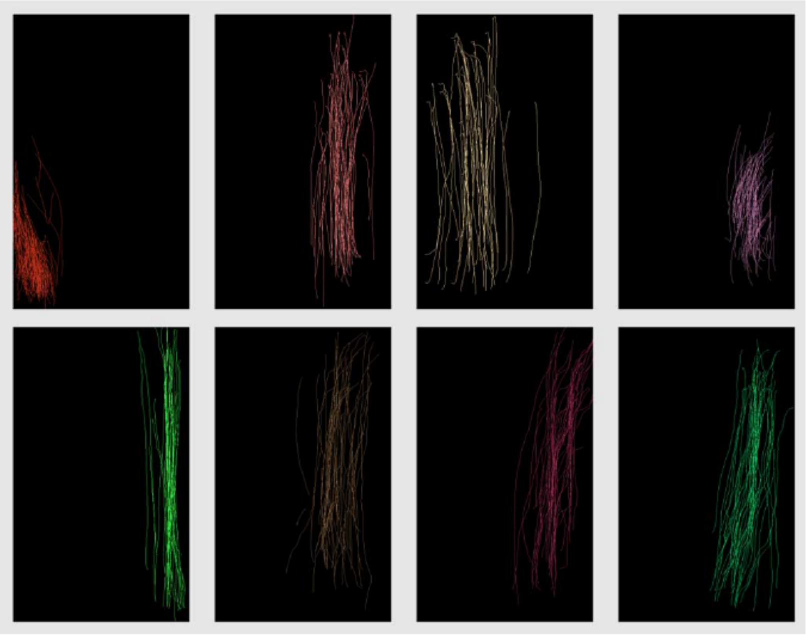


Figure 4.3. Touch Behaviors When Touching a Smart Device Screen (Frank et al., 2013)

a user's finger touches the device's screen and moves it around until it is lifted off. Therefore, each CT generates a special trajectory sequence of touched points (P)s. There are many Ps for one CT created by a particulate user. This led the study to indicate different ids (identifications) for every P since one CT contains many points. Moreover, each P has the characteristics of the touch-screen fusion data features. Figure 4.4 shows the CT illustration created by a user's touches on a screen with its points.

The application captured the single-touch-screen features from UI# 4 and 6. The single-touch-screen sensor has a total of six features. The touch-screen actions (F4) indicated the finger action on the screen such as 'up,' 'down,' 'cancel,' and 'move' action types. Each of the touch-screen actions associated with the touch position of x-coordinate (F5) and y-coordinate (F6) in pixels. The touch-screen actions also associated with the pressure (F7) and size (F8) of the area covered by the touching finger, that have a range value from zero to one. Each of the touch-screen actions has time-stamp (F9) in milliseconds (Frank et al.,

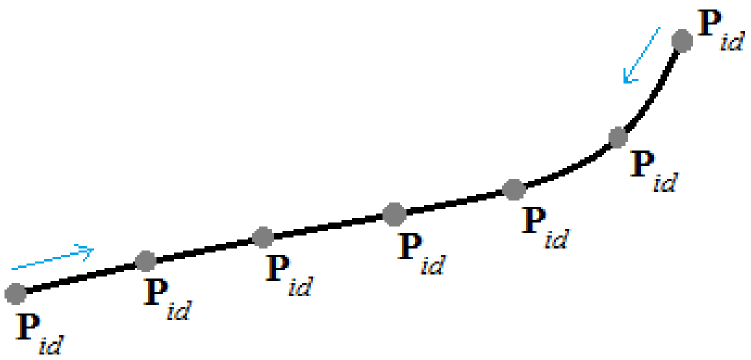


Figure 4.4. Curve Touch When Touching a Smart Device Screen

2013).

An example of touch-screen features that captured the data entries during the recording process on the application are in Table 4.3. The example table shows whether the single touch actions of the finger over the screen were ‘up,’ ‘down,’ ‘cancel,’ and ‘move’ action types. If the user touches the device’s screen, the application stores a ’Down’ action. If the user lifts the finger off the screen, the application stores an ’Up’ action. The ’Cancel’ action occurs when the finger action gets interrupted for any reason. If the user keeps the finger down and moves it around the screen from one position to another, the application captures that as a ’Move’ action.

### 4.2.3. Multi-touch-screen Sensor

The application also captured the multi-touch-screen features from UI# 5. The multi-touch-screen features are similar to the single-touch-screen features except that users are using two fingers instead of using only a single finger to zoom in/out of the picture that was provided in the application task to answer the related picture questions. The multi-touch-screen sensor has a total of ten features. The multi-touch-screen features has one feature for the actions (F10) and the time-stamp (F19) in milliseconds. It has two features for each of

Table 4.3. Touch Screen Feature Data Structure

| T_Actions | XCoor | YCoor | Pr | S | T_T |
|---|---|---|---|---|---|
| Down | 157.0 | 284.0 | 0.1425 | 0.1311 | 34450086 |
| Move | 318.0 | 246.0 | 0.6151 | 0.1231 | 34450162 |
| Move | 337.2 | 290.0 | 0.6632 | 0.110 | 34450208 |
| Move | 348.0 | 306.6 | 0.6121 | 0.1312 | 34450309 |
| Cancel | 349.0 | 310.1 | 0.6781 | 0.0636 | 34450316 |
| Down | 357.0 | 315.2 | 0.7511 | 0.1311 | 34450354 |
| Move | 366.0 | 325.0 | 0.7536 | 0.6131 | 34450377 |
| Move | 366.0 | 325.0 | 0.7563 | 0.6101 | 34450460 |
| Move | 66.0 | 609.0 | 0.4612 | 0.1471 | 34450566 |
| Up | 66.0 | 609.0 | 0.4611 | 0.1414 | 34451205 |
| ... | ... | ... | ... | ... | ... |

the x-coordinate (F11 and F12), y-coordinate (F13 and F14), pressure (F15 and F16), and size (F17 and F18).

### 4.2.4. Accelerometer Sensor

The human hand is a very complex organ. It has many complex and biological and structural features. This research focuses specifically on the holding position or posture of the hand. Every human has a behavioral holding position that differs from others. Therefore, the unique behavioral holding patterns of users may be identified and captured. The application captured these recognition behavioral holding patterns by using the application accelerometer's data. The application accelerometer's data presents the behavioral biometrics of a user's hand whether moving, holding, or orienting a particular application (Muaaz and Nickel, 2012; Nixon et al., 2013b). Figure 4.5 shows the user's holding posture during normal usage (Nixon et al., 2013b).

The application captured the accelerometer features from UI# 1, 6, and 7. The accelerometer sensor has a total of four features. The accelerometer sensor data features come

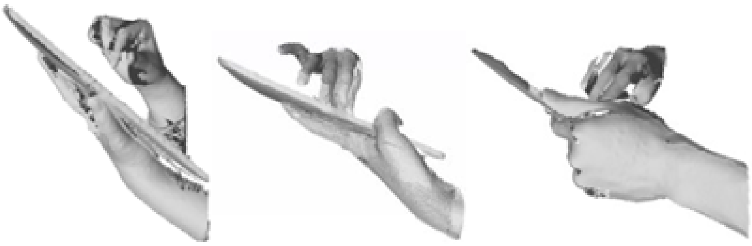


Figure 4.5. Holding posture during normal use of an IPAD (Nixon et al., 2013b)

from three-axes. Those axis measure the changes and the impulses that occur when the device moves in the three dimensions. The x-accelerometer (F20) axis indicates horizontal movement to the right. The y-accelerometer (F21) axis indicates vertical movement upwards. The z-accelerometer (F22) axis indicates movement to the outside of the front face of the device's screen Meier (2012). For every device motion, the application records the time-stamp (F23) in milliseconds. Furthermore, the application not only extracts accelerometer features from reorienting the device as in UI# 7, but it also extracts accelerometer features from UIs that depend on keystrokes and single touch-screen exercises as in UI# 1 and 6. Figure 4.6 shows the accelerometer's three directions of a smart device.

An example of the captured accelerometer features are in Table 4.4. The example table shows the three X, Y, and Z data feature dimension vibrations associated with the time of a user's arm moving the device. The time changes with the three data feature dimensions. The data feature dimension vibrations capture a positive values when the device is in the up position and negative values when the device is in a down position.

## 4.3. Additional Application Characteristics

The application has additional characteristics that help our study to store, hide, secure, and manage the user collected data. All of the user gesture data were recorded in the application's background. The collected data were stored internally in a database using the

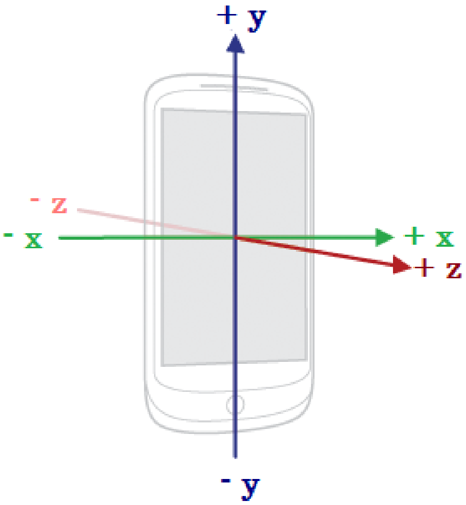


Figure 4.6. Accelerometer Device With X, Y, and Z Dimensions Grenga (2014)

Table 4.4. Accelerometer Feature Data Structure

| **X** | **Y** | **Z** | **Ori_T** |
|---|---|---|---|
| 0.17842 | 0.35681 | 9.02457 | 34450300 |
| -0.12418 | 0.36789 | 9.08632 | 34450302 |
| -0.36742 | 0.46678 | 9.08974 | 34450304 |
| 0.40842 | 0.59008 | 9.14589 | 34450307 |
| 1.00846 | 6.65234 | 0.18907 | 34450308 |
| 1.16543 | 7.79073 | 0.24657 | 34450311 |
| 1.29876 | 8.80932 | 0.35679 | 34450314 |
| ... | ... | ... | ... |

SQLite application (Owens and Allen, 2010). In order to have control over the collected data, we needed to design special UIs that are completely hidden from the users during the data collection process. Access to the data collections UI required a password. These UIs provided us with the opportunity to manage the collected data. Some features of the data collector UIs included the ability to delete the user table, view the user data, and view the number of data samples for a specific table. More importantly, the data collector was enabled to extract the database files from the internal device SQLite database to write to the device's SD card.

### 4.4. Feature Relationships

Two types of relationships exist among the features: Independent and Dependent. Independent features are those that may have its feature freely chosen without considering the features of other features. Dependent features are those that depend on one or more other features. Each feature is associated with its java programming data type, and its variable type. Table 4.5 shows relationship, data ,and variable types between the sensor extracted features. For example in the keystroke features, F1 has an independent relationship with F2 and F3. F1 has a java data type of 'text' and it is a categorical variable. The table excluded the multi-touch-screen features because they provide the same relationship information as the single-touch-screen features.

### 4.5. Sensor Data Inputs

As mentioned earlier, our study obtained data from multiple sensors that generate sets of data features obtained from using the several UIs. Table 4.6 contains the sensor and the number of features extracted as well as the number of UIs that were used by the application. In this table, the feature total is the total number of features per sensor ($\#offeature \times \#ofUI$). By summing up the feature totals, our study utilized 43 features in total.

Table 4.5. Feature Relationships and Types

| **F#** | **Feature Relationship** | **Data Type** | **Variable Type** |
|---|---|---|---|
| **F1** | Independent | Text | Categorical |
| **F2** | Dependent | Varchar | Continuous |
| **F3** | Dependent | Date | Continuous |
| **F4** | Independent | Text | Categorical |
| **F5** | Dependent | Float | Continuous |
| **F6** | Dependent | Float | Continuous |
| **F7** | Dependent | Float | Continuous |
| **F8** | Dependent | Float | Continuous |
| **F9** | Dependent | Date | Continuous |
| **F20** | Independent | Float | Continuous |
| **F21** | Independent | Float | Continuous |
| **F22** | Independent | Float | Continuous |
| **F23** | Dependent | Date | Continuous |

Table 4.6. Extracted Features Per Sensor

| **Sensor** | **# of feature** | **# of UI** | **Feature Total** |
|---|---|---|---|
| **Keystroke** | 3 | 3 | 9 |
| **Single-Screen-Touch** | 6 | 2 | 12 |
| **Multi-Screen-Touch** | 10 | 1 | 10 |
| **Accelerometer** | 4 | 3 | 12 |

# Chapter 5

# DATA ACQUISITION

This chapter concentrates on the data acquisition process and considerations. The chapter starts with the data acquisition process. Next, it discusses considerations of data acquisition such as the device tool, selection criterion, timing period, and design decisions.

## 5.1. The Data Acquisition Process

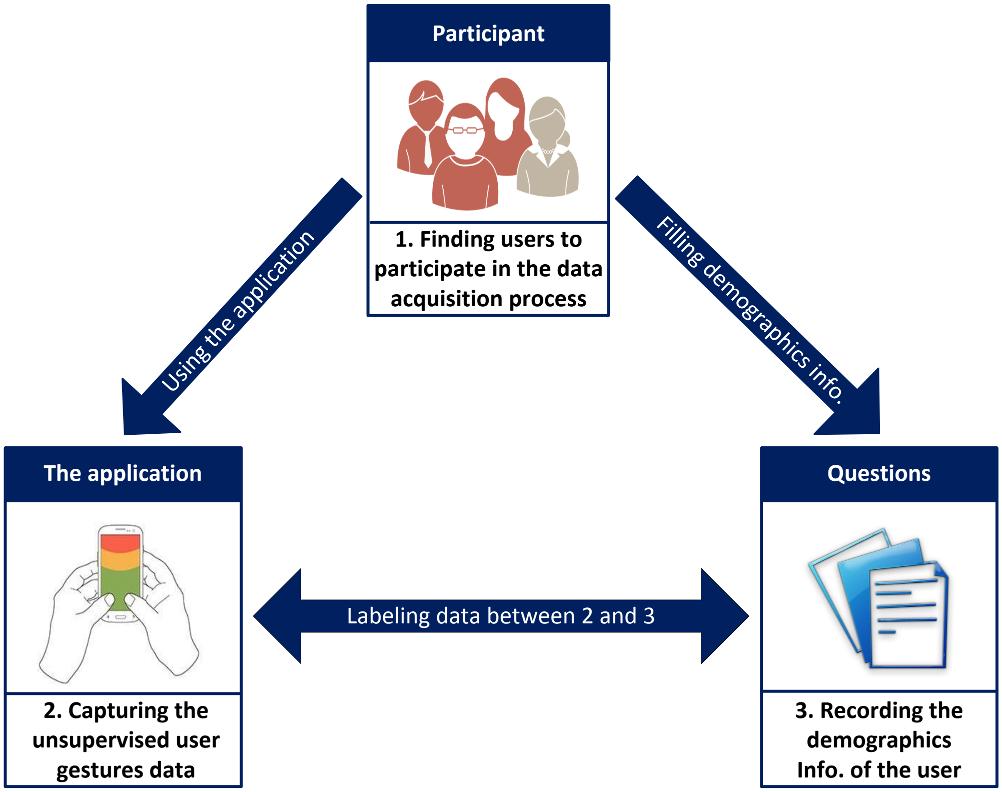

Participant
1. Finding users to participate in the data acquisition process
Using the application
Filling demographics info.
The application
2. Capturing the unsupervised user gestures data
Labeling data between 2 and 3
Questions
3. Recording the demographics Info. of the user

Figure 5.1. Data Acquisition Process Overview

The data acquisition process for our study is illustrated in Figure 5.1. Figure 5.1 illustrates how the participants' application data and demographic group information are gathered. Within this Figure, the application is used to collect unsupervised data from our case subjects comprised of university students and faculty members. The case subjects were asked to participate in an experiment based on "using an Android application to win a free gift card" through fliers posted around the university campus and through word-of-mouth. Each participant received a free gift card as an incentive to participate in the experiment. During the data collection exercise, the data collector assigned an individual user identification number (User-ID) to each participant. The User-ID is used to correlate the data collected from the participant's use of the application and their answers to the consent form that included explicit demographic data considered as the "truth" data. The participants were under the impression that they were simply evaluating the usability of the device and were informed of the true nature of the study after the data was collected. Demographic truth data were obtained from a consent form that each subject completed prior to using the data collection application. The data collection phase was approved by an Institutional Review Board (IRB) for human subject experimentation at the Southern Methodist University Research Compliance (SMURC) with several agreement terms. The description of the agreement terms and the approval copy of the IRB committee are in Appendix B. Hence, the subjects had the option to withdraw from the study after they were informed of its true purpose. In order to enable the supervised learning approach, every User-ID label is included in the application data and the truth data. The consent form included questions as shown in Table 5.1.

## 5.2. The Data Acquisition Considerations

### 5.2.1. The Captured Device Tool

Table 5.1. Subject Questions to Obtain Truth Data

| D# | The question |
|---|---|
| **D1** | What is your gender? |
| **D2** | What is your native language? |
| **D3** | What type of operating systems do you use? |
| **D4** | What is your nationality? |
| **D5** | What is your age? |
| **D6** | What is your social status? |
| **D7** | What is the highest education level you have completed? |
| **D8** | Are you left-, right-, or both-handed? |
| **D9** | How many times do use your device to read emails per day? |
| **D10** | How many times do use your device to view pictures per day? |
| **D11** | How many times do use your device to play games per day? |

All the data were collected from a single smart device, the ASUS MeMO Pad Tablet (ME173X) (ASUS), to avoid biases among multiple devices. The main reason for choosing a tablet device because was due to its large screen which allowed all the participants with different degrees of touch-pad experience to view and interact with the soft keyboard and the questions clearly. Furthermore, Table A.1 in Appendix A provides detailed information of the device and Figure A.1 shows a picture of the device.

### 5.2.2. Selection Criterion

The criterion that was used in the selection process was to divide case subjects into demographic groups. In order to have a variety of user data, our project attempted to select different case subject groups that do not share the same demographic characteristics.

### 5.2.3. Timing Period

The work estimated the overall time that it took a case subject to interact with the application. The estimated time was calculated as: two to three minutes for the seven UI options, thirty seconds for the Log-in UI, and thirty seconds for the final UI. Additionally, all of the volunteers had to fill out and read the two short on-line forms which took around three minutes. The period of time for each data collection exercise was approximately 25 to 30 minutes per case subject.

### 5.2.4. Design Decisions

The study designed a set of guidelines to ensure that all participants had nearly the same environmental set up during the collection phase. These guidelines were:

- None of the participants were aware of any of the application tasks.
- All of the participants must use the same UIs in the application.
- The application allowed the participants to select any task at any given time in no set order.
- The participants were required to hold the device in a steady manner while seated in a chair because the accelerometer sensor is very sensitive with respect to excessive movement.
- The application text font size was fixed and the questions were clear to see.
- The application screen brightness, keyboard type, and UI background colors were fixed for all participants.
- The application layout orientation is fixed to be only in vertical layout mode to ensure that all participants have the same layout mode and to avoid biases by potentially using the device in a horizontal mode.

# Chapter 6

# METHODOLOGY

Our methodology was implemented using the *R* statistical and machine learning software tool set. *R* is an open-source programming language that has several ML algorithmic packages and libraries (R Development Core Team, 2008).

## 6.1. Data Integration

The *R* programming language is a powerful tool that may be used to connect, extract, and modify data in any structured query language (SQL) database. The study took advantage of using *R* to extract the SQL data tables from the SD card of the device into the *R* space. Our primary objective was to integrate all of the tables into one large data table per user, that are subsequently used as the MF tables. The integration process was performed to gather all of the columns of the biometric tables from the keystroke, screen-touch, and accelerometer data sensors and to organize them such that they were adjacent to one another. This process was also performed to access each row of user data from the biometric tables based on the User-ID field as explained in Chapter 5. After selecting data for each user from each of the biometric tables, we combine the columns of feature data and save the produced user table separately. This step was repeated until all user data were processed. Finally, each set of user data has separate MF data files that contain 43 features as previously explained in subsection 4.5. For conciseness, we rename the features to more succinct names such as F1, F2, F3, and so on. Table 6.1 illustrates an example of an MF data frame containing all features associated with a particular User-ID. The Table contains some missing values, that are represented with a value of ”not available” (NA) due to the vector lengths of the features being different in size.

Table 6.1. MF Data Table Example after Implementing the Data Integration Process

| **F1** | **F2** | **F3** | **F10** | **F11** | **F12** | **F13** | **F14** | **F15** | **F40** | **F41** | **F42** | **F43** | **User-ID** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| On | 49 | 34288405 | Down | 157.0 | 284.0 | 0.1425 | 0.1311 | 34450086 | 0.17842 | 0.35681 | 9.02457 | 34450300 | 2 |
| After | 49 | 34288503 | Move | 318.0 | 246.0 | 0.6151 | 0.1231 | 34450162 | -0.12418 | 0.36789 | 9.08632 | 34450302 | 2 |
| Before | 49 | 34288611 | Move | 337.2 | 290.0 | 0.6632 | 0.1101 | 34450208 | -0.36742 | 0.46678 | 9.08974 | 34450304 | 2 |
| NA | NA | NA | Cancel | 349.0 | 310.1 | 0.6781 | 0.0636 | 34450316 | 1.00846 | 6.65234 | 0.18907 | 34450308 | 2 |
| NA | NA | NA | NA | NA | NA | NA | NA | NA | 1.16543 | 7.79073 | 0.24657 | 34450311 | 2 |
| On | 101 | 34288660 | Down | 357.0 | 315.2 | 0.7511 | 0.1311 | 34450354 | 1.29860 | 8.80920 | 0.35650 | 34450318 | 3 |
| After | 101 | 34288707 | Move | 366.0 | 325.0 | 0.7536 | 0.6131 | 34450377 | 1.29790 | 8.80930 | 0.35650 | 34450322 | 3 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

## 6.2. Data Pre-processing

The data pre-processing techniques involve several operations that ensure the uniqueness of the user data. We implemented these techniques on each set of user-collected data separately through the following steps:

1. Convert To Numerical: We convert some of the categorical features in the keystrokes and touch-screen to simplify and equalize them with other sets of numerical features (Alharbi and Thornton, 2015).

2. Missing Values: Our data features did not contain a large number of missing values. The missing values only appear in the multi-touch-screen feature set because some users did not use both fingers to zoom in/out on the displayed picture in the application that is based on displaying a picture. Overall, the percentage of missing data is less than 3.1% of the data per user. Our solution was to replace the missing values with zero.

3. Constant Values: This step is an optional step. Accelerometer features have the limitation of containing some constant bias effects. The solution was to subtract the mean from each value in the accelerometer feature data.

4. Smoothing: This step is an optional step. Accelerometer features may have some extreme noise values (i.e., spikes) because users may quickly move the device unintentionally. The solution was to apply a weighted moving average (WMA) filter with a small sliding window of size five to overcome this noise problem (Muaaz and Nickel, 2012). The noise data were removed after applying this filter. The removed data was less than 1% of the total amount of data per user. Figure 6.1 shows an example of x-accelerometer feature data before (in blue) and after WMA filtering (in red) for an example set of user data.

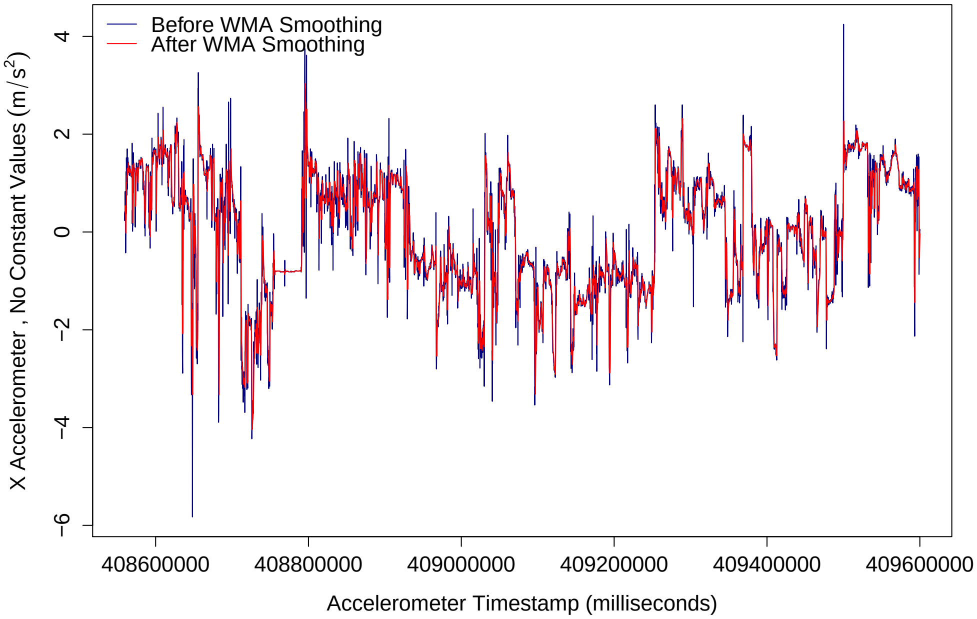


Figure 6.1. Before and After Applying WMA Filter

5. Normalization: This step is used to normalize all the features so that they conform to the same scale of distribution. This step used the standard z-score transformation, which subtracts the feature means and then divides them by the feature's standard deviations to have an average of zero and a standard deviation of one (Alharbi and Thornton, 2015). This step is an important step before applying any data reduction

method because it normalizes the data variance values.

6. Scaling: All features are initially collected in different scale ranges. We re-scaled all data to lie in the range of zero to one in order to avoid biasing problems and to avoid exceeding unacceptably large values (Lin et al., 2013). Figure 6.2 shows an example of feature data before applying scaling filter on the left side and after applying the filter on the right side.

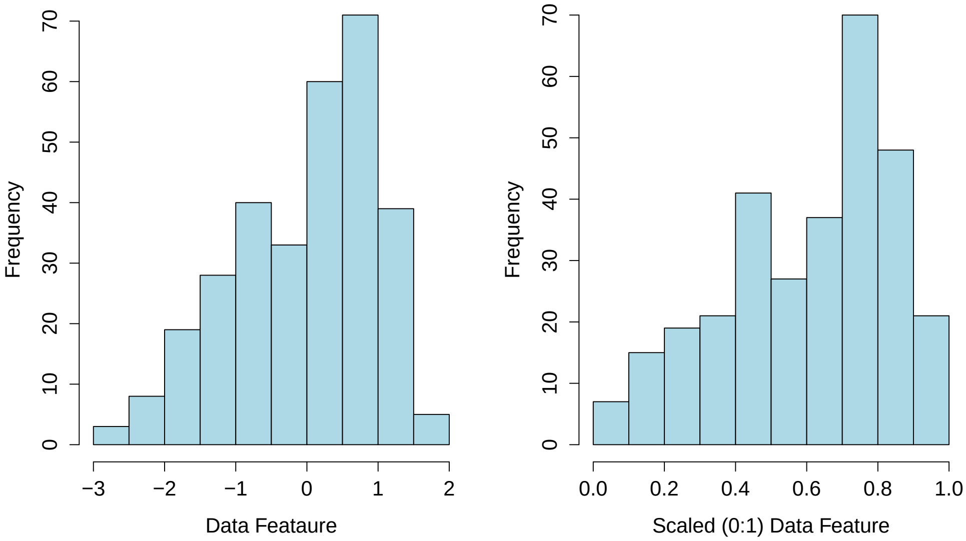


Figure 6.2. Before and After Applying Scaling Filter

7. Equalization: The feature data have different data sample lengths because they are generated from different UIs. In order to fuse them all together, we added zero values the subsets with fewer recorded values such that the total sample sizes would all be equivalent to the maximum number of collected data points among the different feature data sets.

8. To transform our unlabeled data to be applicable for a supervised ML approach, we labeled them with the user demographic group classes using a look-up method. We

used the User-ID as a linking key field between the two the data sets as explained previously in Figure 5.1.

## 6.3. Data Reduction

Smart devices have data storage limitations. The research target is to design a feasible solution with limited data requirements. The collected data must be as small as possible in order to be stored in the device's memory or the cloud. The use of reduced amounts of data improves the supervised classification model performance without losing useful information (de Mendizabal-Vzquez et al., 2014). Furthermore, restricting the amount of data is important if a cloud-based solution is deployed since minimizing the amount of required data to transmit from the device to the cloud will improve timing performance. We chose to use two data reduction techniques, which are:

### 6.3.1. Principal Component Analysis (PCA)

PCA is a technique that has been used to reduce the complexity of MF data and to represent the data with a smaller number of dimensions (Fodor, 2002). PCA uses an orthogonal transformation to identify hidden correlated features and to then transform the data into a new set of values of linearly uncorrelated attributes (principal components). It is also a technique that removes redundancy by discounting data that is not significant by retaining only the most significant principal components. The most significant components are chosen based on variance values of the MF data (Duda et al., 2012; Panahi et al., 2011).

By implementing the PCA technique in our study, we had the ability to compute forty-three new principal components and rank them in top-down order based on the component variances. It also allowed us to select the first 15 highest-ranked components and to disregard the 28 lesser-ranked components that have low variance values that contain a rate of change of zero. We considered 0.3% of principal component percentage variances as a cut off threshold value. Therefore, we kept any principal component that has above this threshold value. It is important to remove the lesser-ranked components because they will affect the classification learning process (Williams, 2009). Figure 6.3 shows the first 20 principal components and the percentage of variances. Component one has 65.6% of the variance, where the second component has less percentage of variance and so on.

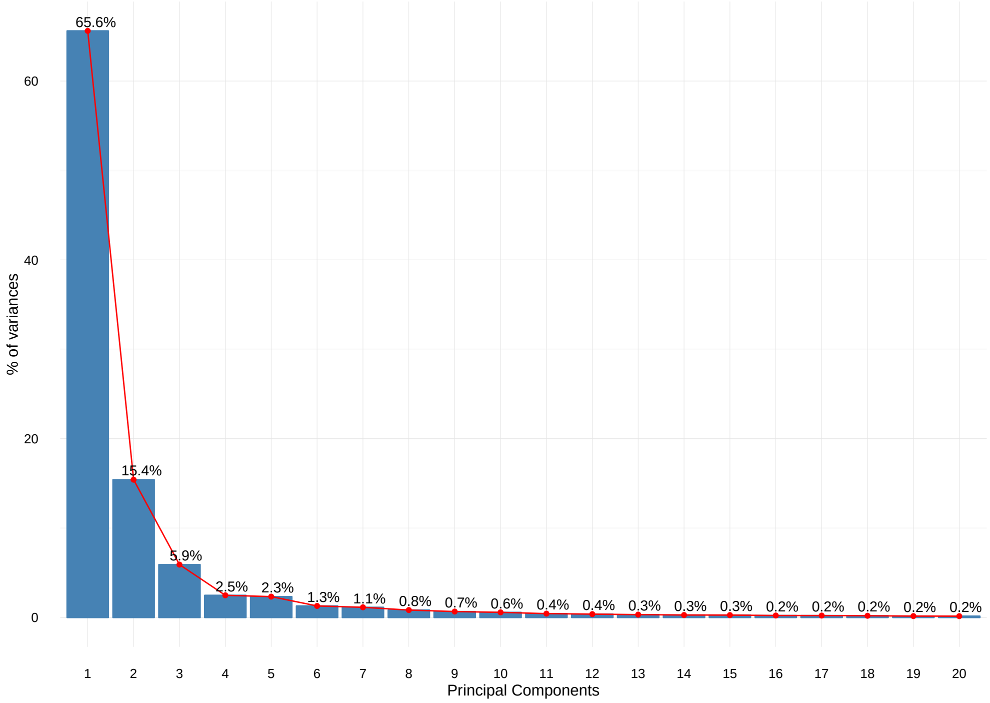


Figure 6.3. Principal Components Versus Percentage of Variances

Similarly, Figure 6.4 shows the first 20 principal components and the eigenvalues. Component one has 2.4 of the eigenvalue, where the second component has less eigenvalue and so on.

Moreover, Figure 6.5 shows the bi-plot that presents the relations between two new principal components and the original MF features. This plot is used to show the proportions of each feature along the two principal components. It shows the most important MF features and gives the relative importance of the two principal components. It takes two principal component observations and underlines over the MF features. Hence, each MF feature has

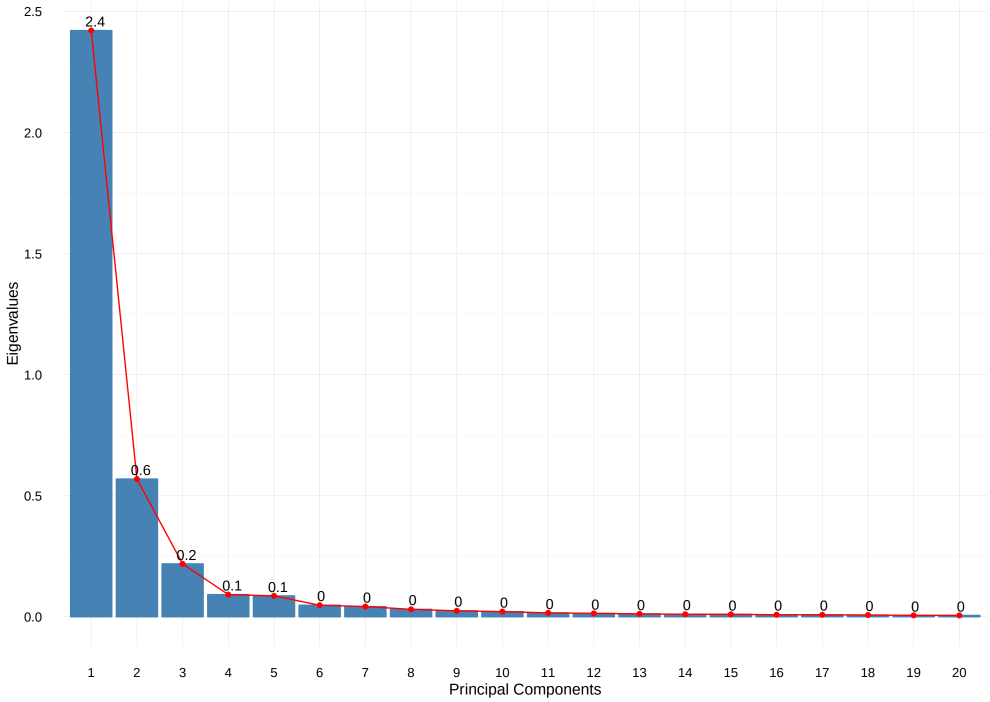


Figure 6.4. Principal Components Versus The Eigenvalues

its direction over the two principal component observations. For example, the F41 feature has the same direction of the F43 feature over the two principal component observations.

### 6.3.2. Self-Organizing Maps (SOM)

SOM is an unsupervised classification technique and it is not a clustering technique. Our study implemented SOM as a data compression technique. SOM is a type of artificial neural network algorithm discovered by Teuvo Kohonen in (Kohonen, 1982). It uses a finite number of artificial neurons to map and reduce a higher dimensional space into a lower dimensional space; typically a two-dimensional (2-D) grid or lattice. SOM organizes nodes or units in a network into two layers, the input layer and the output (Kohonen) layer. Within the output layer, each node is fully connected to the input layer and there is no connection between the input nodes. Every node contains a vector of weights present on the links to a neuron of the

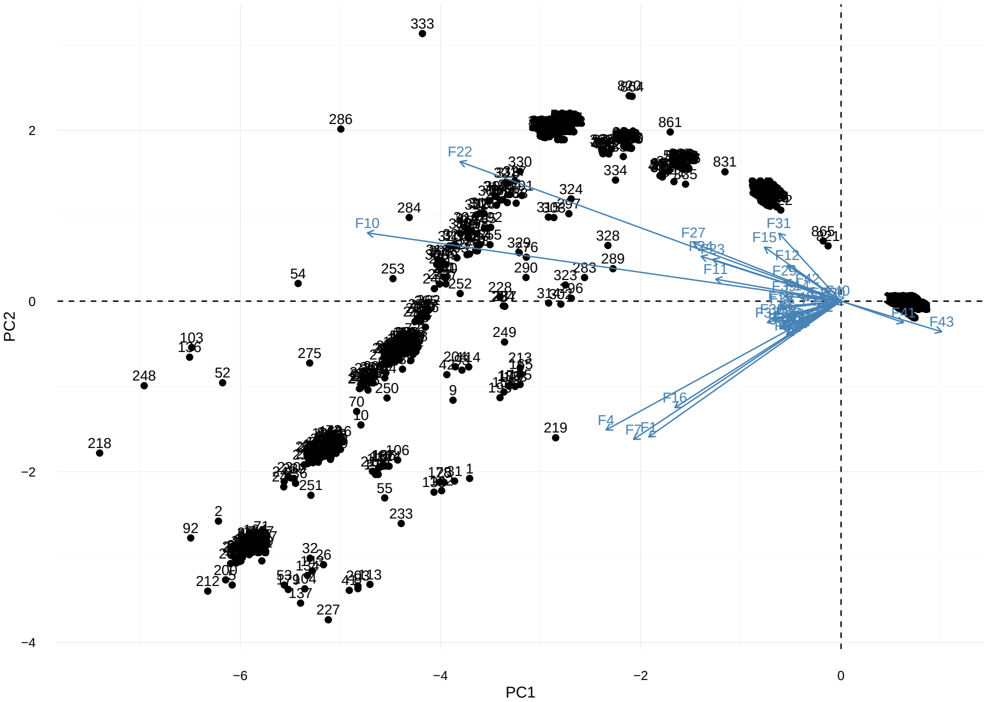


Figure 6.5. MF Features Over Two of The Principal Component Observations

same dimension as the input vectors. These weights represent the neuron output vector and there is neighboring relationship among the neurons. There are advantages to implementing the SOM technique after PCA is applied. PCA only reduces the MF dimensions to a few principal components, whereas SOM reduces and addresses the issue of having a different number of observations among the principal component vectors by outputting and equal number of samples. The result is that each user data set, after PCA and SOM processing, is comprised of the same number of dimensions and samples. The SOM technique is also used to determine previously undetectable or "invisible" patterns, remove noise data, and keep the neighborhood relationship among the data in its mapping representation. Briefly, the SOM algorithm operates according to the following steps:

1. Select the 2-D grid size and shape type. The size is defined as $M \times M$, where $M$ is the

number of columns and rows of the grid (i.e. 10 × 10). Typically, the shape is either hexagonal or square shape types.

2. Initialize all node weight vectors with a random value between 0 and 1.

3. Choose a random input data point from the training data.

4. Find the best matching unit (BMU) based upon the weights and data points by calculating the most similar node in the map using a Euclidean distance formula to enable a similarity measurement.

5. Determine the size of the neighborhood for the nodes around the BMU and wherein that size is exponentially decreased with each iteration.

6. Adjust weights of each node in the BMU and neighboring nodes to become more similar to the weight of the chosen data point. This causes the learning rate to decrease with each iteration. As training continues, the neighborhood progressively shrinks until it reaches a size of zero.

7. Repeat steps 2 to 5 for $N$ iterations.

Our motivation for implementing the SOM technique was to select different grid sizes and we intuitively chose the hexagonal mapping shape type because each node would have more immediate neighbors. In order to find the required number of training iterations, we calculate the distance from each unit using weights of the samples represented by the unit as it is decreased. Figure 6.6 shows the curves of the training progress over number of iterations for four selected grid sizes in $10 \times 10$, $20 \times 20$, $30 \times 30$, and $40 \times 40$. For every curve, the mean distance to closest unit is continually decreasing as more as iterations are processed. From this figure, it is apparent that any number of iterations above 400 has a low mean distance to closest unit. Our study chose a number of iterations equal to 500. We used the Kohonen library in the implementation described here (Wehrens et al., 2007). A detailed explanation of this technique can be found in (Guthikonda, 2005; Yin, 2008).

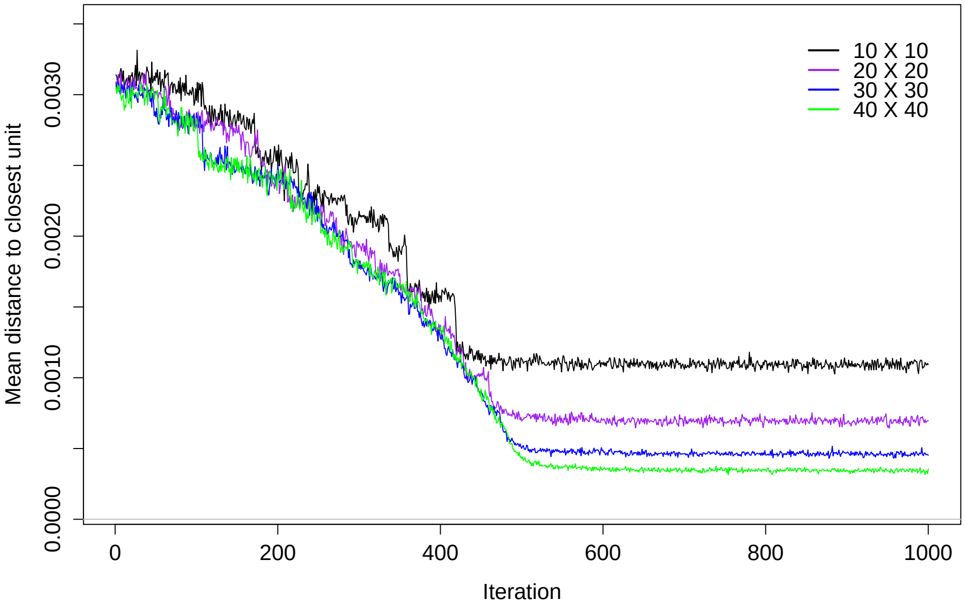


Figure 6.6. Mean Distance to Closest Unit Over Iterations

We depict a user heat-map plot to observe the distribution of observations across the map as shown in Figure 6.7. The Figure shows each node in a cell that is color-coded. The color code legend is shown on the right side of the plot. It also shows the similarity of one particular cell to all other cells in the map allowing one to observe the mean similarity of all cells. For example, the light blue cells in top left corner of the map belong to one cluster because they share the same mean similarity.

The primary reason we chose both the PCA and SOM techniques was due to related studies (Silva and Del-Moral-Hernandez, 2011; Wu and Li, 2014) that indicated that PCA+SOM improve classification method performance. Many past classification algorithms also have used these techniques to design specific classification systems that are used in the biometric applications. In our case, PCA and SOM techniques are used to discover the hidden patterns and reduce dimensions and observations than the original data. Both techniques perform as

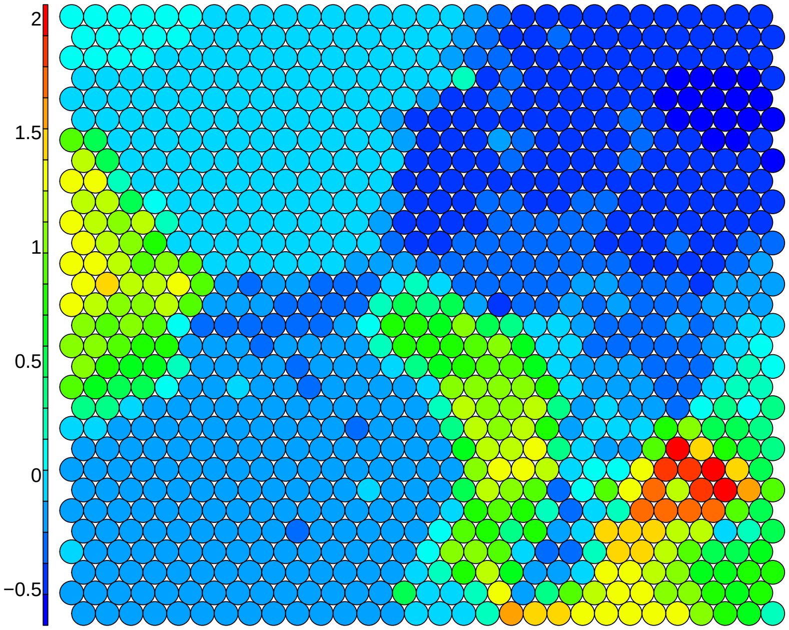


Figure 6.7. SOM Heat-Map Plot After Implementing PCA

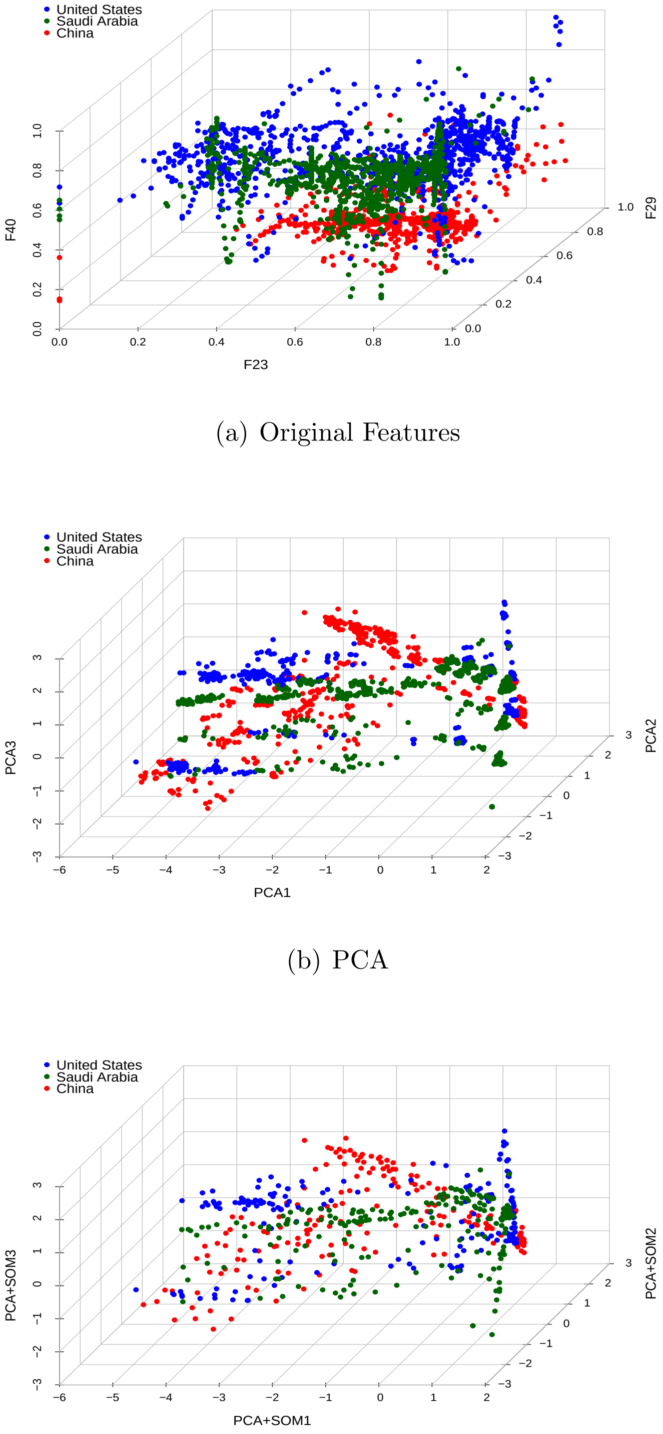
an unsupervised classification (ignores class labels). The PCA technique extracts the significant variance dimensions, where the SOM technique find the mean similarity between the dimension observations. In order to illustrate the impact of implementing the PCA+SOM technique in our MF data space, we visualize them using a three-dimensional (3-D) scatter plot. We picked three user nationalities: United States, Saudi Arabia, and China. Each nationality is represented with different colors: blue, dark green, and red, respectively. For simplicity and clearness, we selected only the first 1000 observations for each user in this graph. Figure 6.8 shows three 3-D scatter plots before implementing PCA and PCA+SOM techniques and after. Plot (a) represents three selected features (F23, F29, and F40) where the points are distributed randomly in the space. Plot (b) presents the three highest principle component dimensions without applying SOM technique, where the points are skewed; however, the number of the observations are still the same. Plot (c) presents the three highest principal component dimensions after applying SOM technique using a $15 \times 15$ grid, where the points are reduced to less than 77.5% of the original observations and are structured based on their similarities.

## 6.4. Supervised Classification

In this section, we compare three different supervised classification models for obtaining the study's objectives motivated by the related studies in section **??**. We implemented the classifiers based on the caret library, which contains a large variety of classifiers (Kuhn, 2008).

### 6.4.1. Artificial Neural Networks (ANN)

Neural networks are widely recognition for use in studies due to their capability to learn relationships among variables. They are a computational approach that creates a collection of neural units or nodes comparable to the way the brain solves problems. There are several versions of ANN, we picked the popular NNET classifier. It consists of neurons that are configured as a layered structural network. It receives input vectors and converts them into output prediction results. Every neuron obtains input data and applies non-linear functions that then transfer the results to the next layer. It is structured as a feed-forward type of network where a neuron outputs its results to other neurons that are in the next layer. In this type of ANN, there is no feedback to the previous layer. Weightings are applied to the signals passing from one neuron to another which are tuned in the training phase to adapt the neural network to the particular recognition problem at hand. It has the general architecture of ANN, which contains a number of inputs and one predicted output. The inputs could be either our MF features directly, or the most significant principal components, while the output is the predicted user demographic. It also has the ability to contain a hidden layer between the input and output layers. We chose the number of nodes for the hidden layer to equal the number of nodes in the input layer. Figure 6.9 shows an example plot of ANN



(a) Original Features

(b) PCA

(c) PCA+SOM

Figure 6.8. Three 3-D Scatter Plots for 3 User Nationalities Before and After Applying PCA and PCA+SOM

classifier. It presents 15 inputs and one hidden layer that contains 15 nodes. The output is the demographic group class in this network.

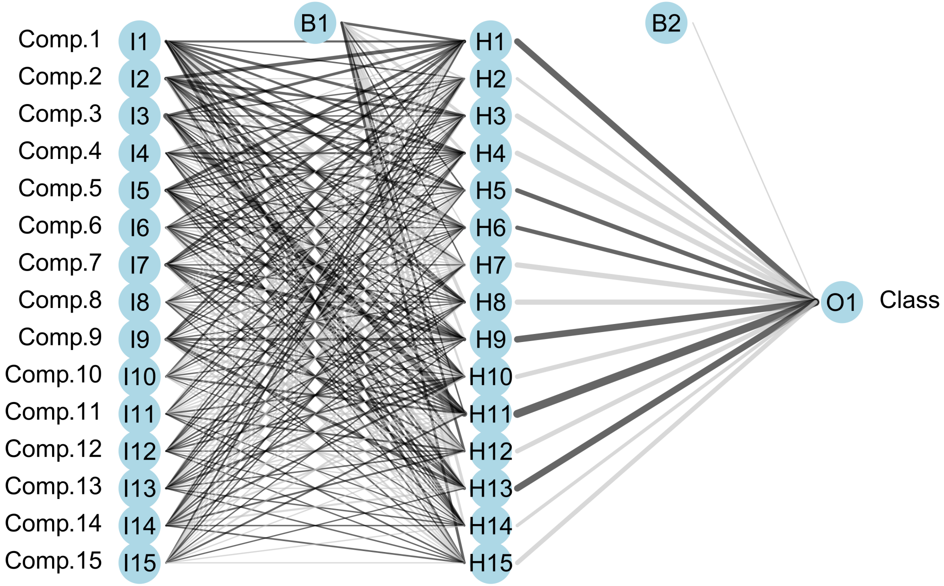


Figure 6.9. Example Plot of ANN Classifier

### 6.4.2. Decision Tree (DT)

There are many types of DT classifiers. We chose J48 (also known as C4.5) to serve the goal of predicting which user data set belongs to a particular set of demographic groups. It also has the ability to perform as a feature selection method. The C4.5 classifier is a pruned tree that is based on a top-down strategy, and a recursive divide and conquer strategy. It spreads feature values and categorizes them into leaves by selecting the feature to split on at the root node. Then, it creates a division for each possible feature value and splits the variables into subdivisions, one for each division that extends from the root node. It repeats the execution recursively for each division by selecting a feature at each node. It uses the only variable that arrives at that division to make the selection. Figure 6.10 shows an example plot of DT classifier. This plot contains nodes, where it structured into top-down strategy. The parent node decides which values belong to the children node. The last nodes in this tree are the leaf nodes.

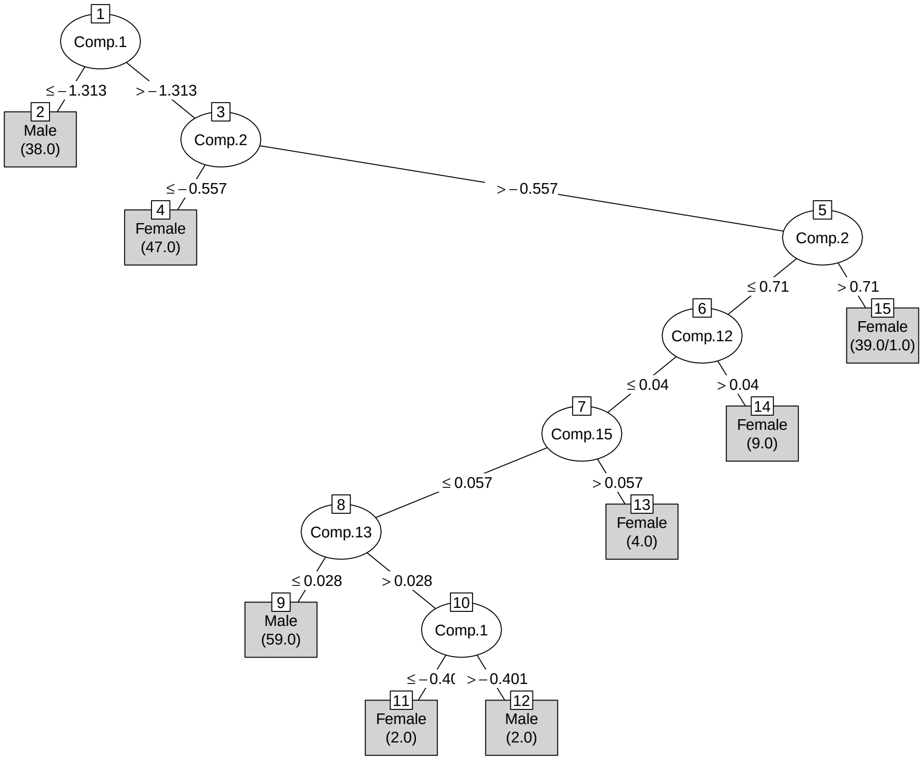


Figure 6.10. Example Plot of DT Classifier

### 6.4.3. $k$-Nearest Neighbor Classifier ($K$NN)

We chose the $K$NN supervised classification model for several reasons. One reason is that it is a robust classifier with respect to the noise that might exist in the feature data since it is based on distance measurements. Second, it provides fast classification and demographic predictions. Finally, it appeared to be a good classifier to be combined with the SOM recognition data. The $K$NN classifier uses the local neighborhood between each new observation (here, the demographic observation) and projects the observation into the feature space with respect to the training observations. The local neighborhood is based on distance measurements that compute the similarity of the nearest observation neighbors. We used the $L_2$ Euclidean distance measurement because it is fast, simple, and yields good performance in many smart device biometric projects de Mendizabal-Vzquez et al. (2014). The $K$NN classifier has the ability to memorize the $K$ training observations that are more similar to the new observations. Hence, it chooses the label that has the majority of the $K$ closest training observations. This classifier, in essence, stores all training observations and labels, which may be a limitation where large data sets are used. In our case, this limitation is not a problem because our methodology initially uses PCA+SOM data dimensionality reduction techniques to store fewer dimensions than those present in the initial demographic observations. In our study, we use a $K$-value equal to the number of the classification problems as discussed in Table 1.1. Figure 6.11 shows an example plot of $K$NN classifier. In this plot, we picked up the education level (D7) classes, where we color coded each class as: bachelor (red), doctoral (green), and master (blue). Between each class there is the decision boundary of the $K$NN classifier over two principal component observations.

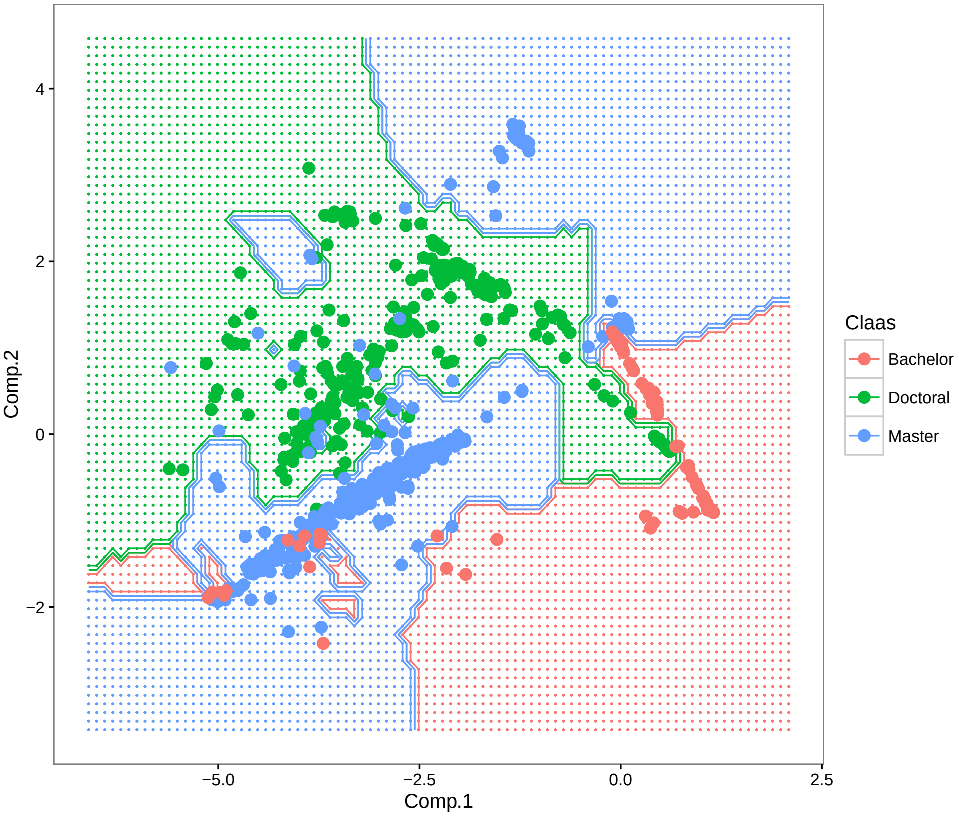


Figure 6.11. Example Plot of $K$NN Classifier

# Chapter 7

# DATA ANALYSIS

The total number of observations collected from our population of test subjects was 437,590. Each participant who used our application generated between 9,722 to 2,043 observations among the 43 features. In this chapter, we focus on providing some basic data analysis on the collected data after applying pre-processing phase.

## 7.1. Data Normality

Our study applied the Shapiro-wilk and Anderson-Darling normality tests (Razali et al., 2011). These tests resulted in $p$-values of less than $2.2e^{-16}$ for all of the features. These very small $p$-values indicated that the data is not normally distributed. Therefore, we reject the null hypothesis of normality.

## 7.2. Data Variances Analysis

The study considered a comparison of the differences between sample means to a scatter of multivariate feature data within the samples by applying the multivariate analysis of variance (MANOVA) statistical testing method (Neideen and Brasel, 2007). MANOVA is related to the analysis of variance (ANOVA). The reason behind using MANOVA method was to test our 43 response features for a specific demographic group. The test depended on evaluation of different statistical traces such as the Hotelling-Lawley, Pillai, Roy, and Wilks. In our case, we applied the test using the four traces for all of the proposed demographic groups. The tests resulted in $p$-values of less than $2.2e^{-16}$ for all the demographic groups. These very small $p$-values indicated that there is a high significant difference between the sample means. Therefore, we reject the null hypothesis that there is no difference in means across the proposed demographic groups.

## 7.3. Data Feature Ranking

Our study used the $f$-selector function that chooses the best combination of features based on information gain theory of a particular feature's entropy (Cheng et al., 2012). The reason for choosing this method rather than other methods is because the weights assigned by other classification methods are different, which might effect the ranking process. The study's primary purpose for using this technique was to sort the features in descending order and discover the best feature combination. Table 7.1 shows the results of using the method for the first five demographics based on 50 users and provides the top ten MF combinations, where the best feature is on the left and the least contributing feature is on the right. For example, in D1 the best feature is F40 and the least contributing feature is F18. In this table, each demographic has its own combination of features. By considering this set of demographics, the accelerometer features (F40, and F42) are ranked as the best because they have the highest data variances among other features.

Table 7.1. Ranking of Features for the First 5 Demographics

| **D#** | **The top ten ranked feature combination** |
|---|---|
| **D1** | F40+F29+F42+F27+F41+F20+F19+F43+F21+F18 |
| **D2** | F42+F40+F29+F24+F23+F19+F21+F18+F16+F17 |
| **D3** | F40+F42+F41+F29+F1+F2+F3+F4+F5+F6 |
| **D4** | F42+F40+F29+F41+F27+F23+F24+F4+F31+F22 |
| **D5** | F40+F41+F42+F29+F27+F24+F20+F23+F22+F31 |

## 7.4. Data Feature Relationships

In order to better understand the MF relationships and to determine which features provide hidden information, we depict the correlation coefficients by using the Pearson parametric correlation method (Neideen and Brasel, 2007). Figure 7.1 shows the correlation coefficients of all pairs of the features in a color-coded plot. White squares indicate that the feature pair is not correlated, blue indicates a positive correlation, and red indicates a negative correlation. On the bottom side of the correlation matrix plot, the legend color scale ranges from -1 to 1 which indicates the correlation coefficient ranges with the corresponding color codes. The darker a color is, the larger the correlation coefficient is between the feature pairs. With this plot, one feature can be observed to be highly correlated with other features. For example, feature F22 is highly correlated with features F23 and F24. This makes sense intuitively because they are all touch-screen based features. In fact, this plot identifies the hidden structures and patterns that could exist between the features, which guided our decision to implement the PCA technique in order to transform the correlated features into sets of values that are linearly uncorrelated dimensions.

## 7.5. Data Feature Visualizations

There are many ML techniques of data visualization; this section uses scatter, box, and histogram plots (Duda et al., 2012).

The study noted the user genders as male (red) and female (blue) from the X and Y coordination features as shown in Figure 7.2.

Upon further investigation of the user demographic groups, the study revealed that users of different ages apply different touch pressures. Figure 7.3 shows the five different user pressure ages presented with boxes. Each box has its mean in a thick line (Duda et al., 2012).

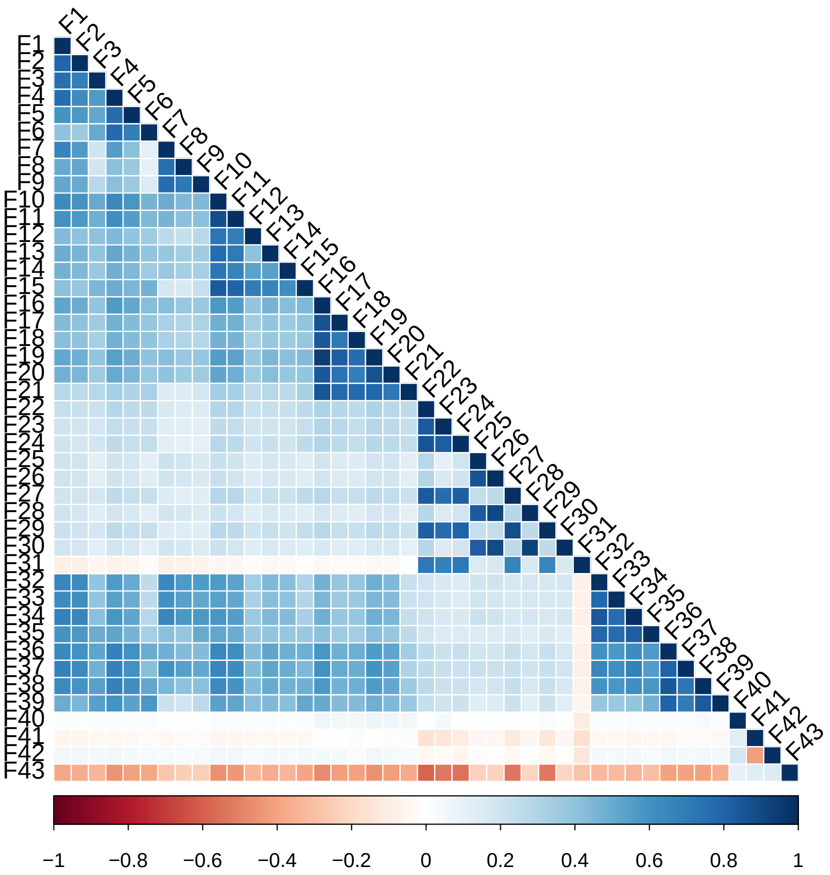


Figure 7.1. Correlation Matrix for all Features

Figure 7.3 shows the 25-year-old user gains of about 0.2 of pressure mean value, while the 28-year-old user obtained the lowest pressure mean value among all the user ages. Another observation between these users' pressure mean was the difficulty they en-counted using the device's application such as the operating system (OS) type (Android vs. other OSs) or trying to use the application itself.

One of the goals was to identify the users' languages. Figure 7.4 shows the keystroke ASCII key code with the frequency, and also shows English users in blue and non-English users in red. The frequency refers to how often the keystrokes might appear. The figure also displays the statistical probability curve for the English keystroke key code when it took

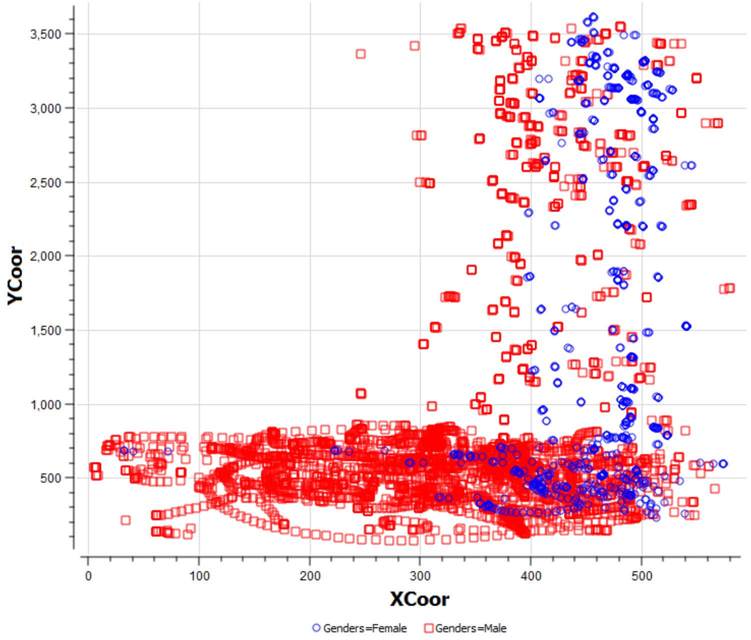


Figure 7.2. User genders by X and Y-coordination features

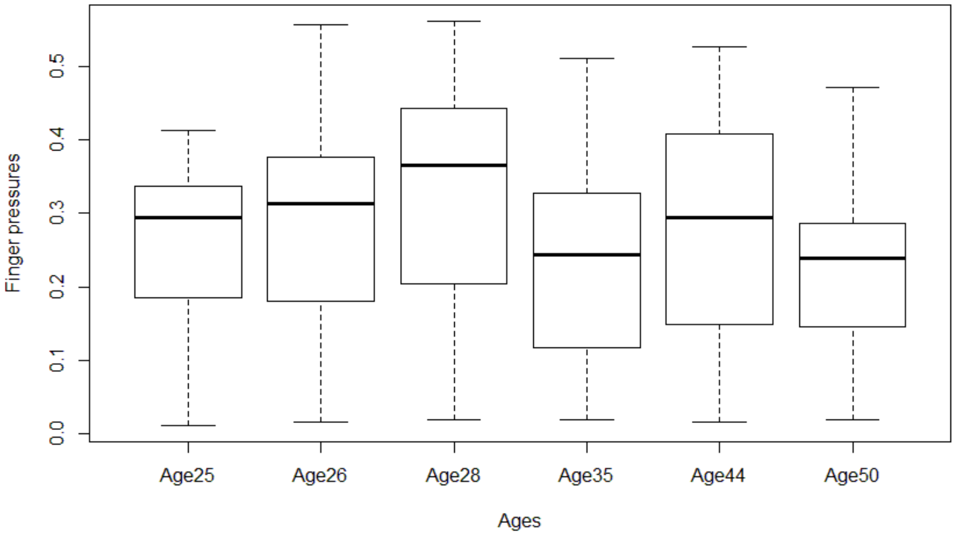


Figure 7.3. User ages by pressure feature

place between 0.1 and 0.2 frequency values (Duda et al., 2012).

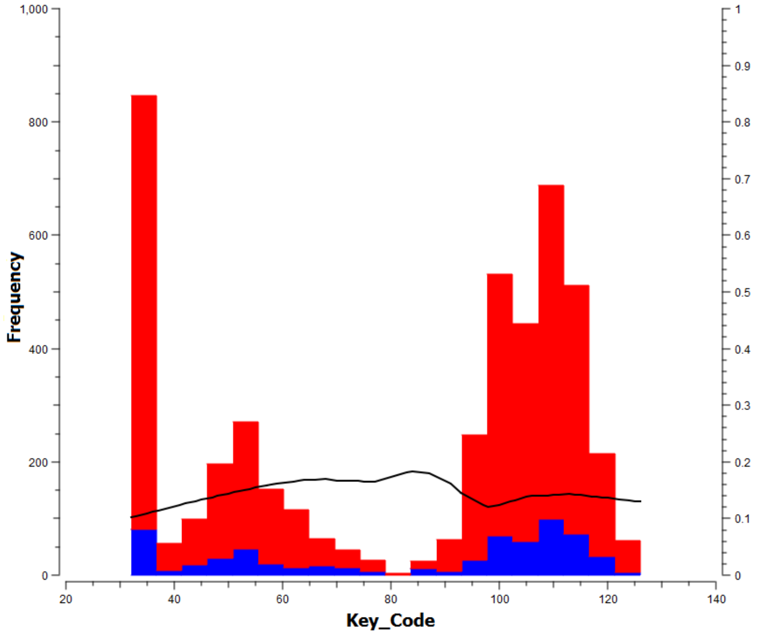
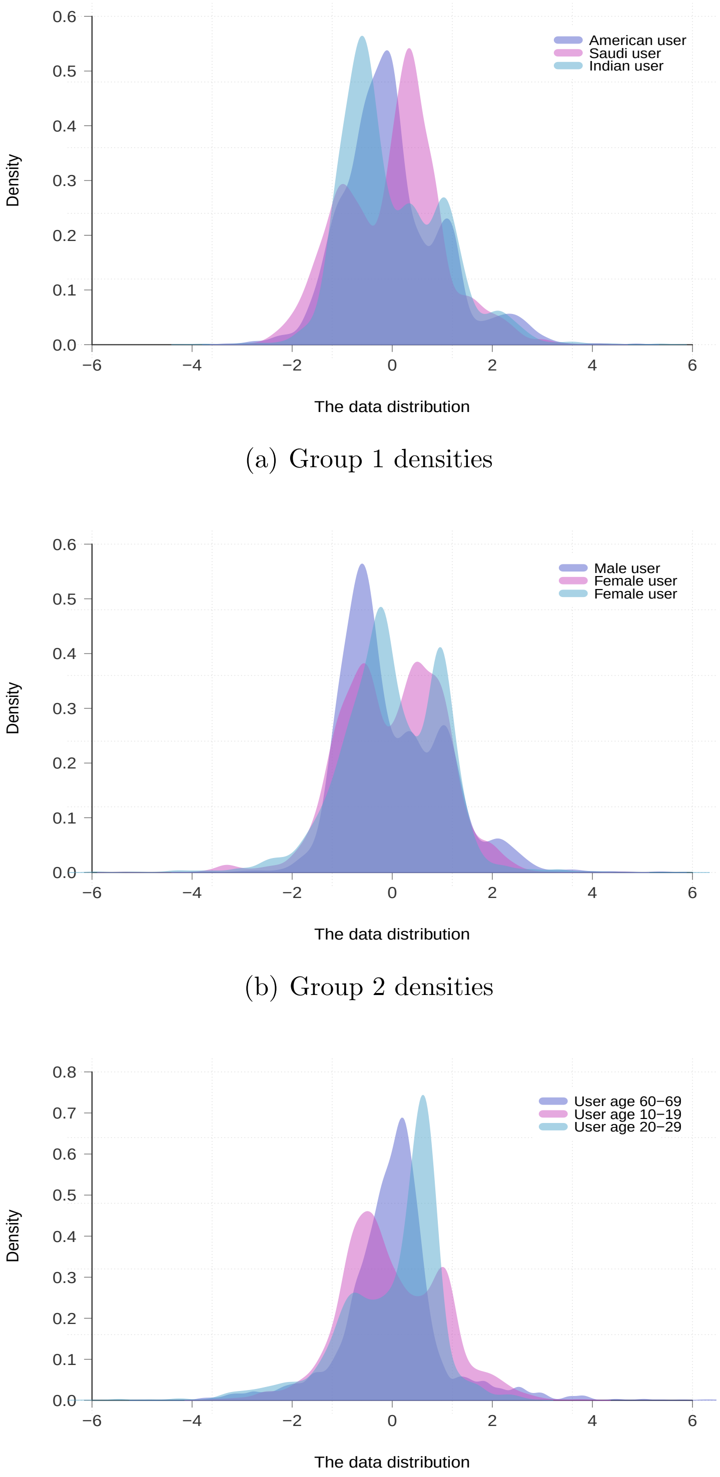


Figure 7.4. User languages by keystroke key code feature

The study visualizes the data distribution by observing the feature density since it gives us the ability to distinguish the user recognition patterns. We pre-selected nine users and divided them into three groups, where each group has three user. Group 1: has users that are sharing the same gender (Males) but different nationalities. Group 2: has users that are sharing the same nationality (Americans) but different genders. Group 3: has users that are sharing the same nationality (Americans) and gender (Males) but different ages. We used the data that generated from the same feature. Figure 7.5 shows the three density plots for three different user groups. From all of these plots, every demographic group of users obtain a special patterns.



(a) Group 1 densities

(b) Group 2 densities

(c) Group 3 densities

Figure 7.5. Three Density Plots for 3 Different User Groups

# Chapter 8

# EXPERIMENTAL RESULTS

This chapter carried out several different experiments for the methodology described here. We used a cross-validation strategy to validate our experimental results. This strategy was based on ten-fold cross validation repeated ten times. All other classifier parameters such as the seed and tune length number were set up under the same situation to avoid biasing with respect to those parameters. The results were based on several classification metrics. The accuracy metric is the percentage of correctly classified observations divided by all observations. The kappa metric is similar to the accuracy metric, but it depends on the base distribution of the classes. Additionally, we obtained the biometric performance metrics. These metrics were based on the receiver operatic characteristic (ROC) curve. The ROC demonstrates the classification model's ability to differentiate between positive and negative classes. The ROC relies on two performance metrics: sensitivity and specificity. Sensitivity measures the number of samples from the positive (first) class that were truly predicted correctly. Specificity measures the number of samples from the negative (second) class that were truly predicted correctly. A perfect classification model that has a high sensitivity, specificity, and ROC metric rates is 100% if the model has made all predictions perfectly. If the model results in rates closer to 50%, then it is no better than a random guess.

## 8.1. Results of Comparison of Different Methods

It is a useful practice to validate the proposed approach after implementing each method in order to observe the accuracy results. We selected four demographic groups (D1, D6, D7, and D9) and ten different sets of user data for this experiment. We validated the classifiers and achieved accuracy rates in situations where the classifiers were implemented without the PCA and SOM techniques for data reduction and also when PCA and SOM techniques were implemented before classification, as shown in Table 8.1. This Table indicates that the PCA technique improves all the classifiers' accuracy rates. The PCA+SOM technique slightly decreases the ANN and DT classifier accuracies and improves the $K$NN classifier accuracy rates.

Table 8.1. Comparison of Method Implementations

| **Method** | **D1** | **D6** | **D7** | **D9** |
|---|---|---|---|---|
| **ANN** | 95.50% | 97.44% | 96.81% | 94.33% |
| **PCA+ANN** | 97.63% | 99.57% | 99.07% | 96.75% |
| **PCA+SOM+ANN** | 94.99% | 99.88% | 96.15% | 89.54% |
| **DT** | 98.53% | 99.39% | 98.52% | 98.07% |
| **PCA+DT** | 99.41% | 99.95% | 99.47% | 99.01% |
| **PCA+SOM+DT** | 97.05% | 99.74% | 96.93% | 95.09% |
| $K$**NN** | 98.92% | 99.50% | 98.84% | 98.22% |
| **PCA+**$K$**NN** | 99.92% | 99.99% | 99.91% | 99.84% |
| **PCA+SOM+**$K$**NN** | 99.61% | 99.86% | 99.51% | 99.00% |

### 8.2. Different PCA+SOM Grid Size Results

In this experiment, we selected four different SOM grid sizes ($10 \times 10$, $20 \times 20$, $30 \times 30$, and $40 \times 40$) after implementing the PCA technique, and applied the three classifiers (ANN, DT, and $K$NN) to classify all the user demographic groups data. Tables 8.2 and 8.3 show the achieved accuracy and kappa rates for each classifier as compared to various selected grid sizes. The classifiers achieve higher rates when the grid size increases. The $K$NN classifier achieved the highest results among the ANN and DT classifiers in every size. Based on the results, the best result of the classifiers are achieved when the grid size is $40 \times 40$. This result guided our decision to implement this size of 40 × 40 for the SOM grid.

Table 8.2. Comparison of Accuracy Rates for Different Classifiers and PCA+SOM Grid Sizes

| D# | 10 × 10 | | | 20 × 20 | | | 30 × 30 | | | 40 × 40 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **ANN** | **DT** | ***K*NN** | **ANN** | **DT** | ***K*NN** | **ANN** | **DT** | ***K*NN** | **ANN** | **DT** | ***K*NN** |
| **D1** | 69.04% | 77.92% | 83.07% | 72.70% | 85.88% | 94.77% | 74.45% | 90.94% | 97.74% | 74.88% | 93.69% | 98.89% |
| **D2** | 72.61% | 79.84% | 85.07% | 76.12% | 87.25% | 94.96% | 77.51% | 91.35% | 97.99% | 77.76% | 93.98% | 99.00% |
| **D3** | 62.87% | 72.99% | 81.42% | 67.68% | 84.70% | 94.13% | 69.33% | 89.78% | 97.49% | 70.10% | 92.59% | 98.75% |
| **D4** | 30.70% | 54.17% | 47.38% | 33.80% | 73.38% | 75.45% | 34.67% | 82.76% | 86.36% | 35.88% | 87.91% | 91.91% |
| **D5** | 75.64% | 79.45% | 84.89% | 77.29% | 87.86% | 93.89% | 78.05% | 92.01% | 97.05% | 78.15% | 94.34% | 98.42% |
| **D6** | 88.40% | 91.24% | 92.83% | 90.70% | 94.87% | 97.57% | 91.42% | 96.55% | 98.94% | 91.69% | 97.49% | 99.45% |
| **D7** | 58.08% | 68.59% | 80.36% | 62.51% | 82.37% | 93.49% | 64.67% | 88.70% | 97.11% | 65.00% | 91.69% | 98.56% |
| **D8** | 70.57% | 77.89% | 84.37% | 73.41% | 86.24% | 94.54% | 74.47% | 90.89% | 97.72% | 75.21% | 93.66% | 98.82% |
| **D9** | 44.37% | 61.13% | 72.27% | 48.55% | 77.53% | 90.42% | 50.57% | 85.43% | 95.59% | 51.70% | 89.69% | 97.73% |
| **D10** | 41.39% | 59.88% | 71.09% | 45.38% | 77.15% | 89.92% | 47.21% | 84.90% | 95.49% | 47.77% | 89.11% | 97.65% |
| **D11** | 35.00% | 57.22% | 68.32% | 39.52% | 75.21% | 88.70% | 41.08% | 83.53% | 94.81% | 41.41% | 88.48% | 97.27% |

### 8.3. Biometric Performance Results

The biometric performance metrics are only fit for binary classification problems. However, our study is concerned with multi-classification problems for some demographics (D4, D5, D7, D8, D9, D10, and D11) as in Table 1.1. Thus, we used only the two highest group percentage classes from those demographics. For example, in D4, the study considered only the United States and India user classes because they comprised the highest number of subject data samples. It is noted that we changed the value of $K$ in the $K$NN classifier to make it binary (two). Table 8.4 shows the high biometric performance rates of the sensitivity, specificity, and ROC for the proposed approach.

Based on the comparison of the performance rates of Tables 8.2, 8.3, and 8.4, the $K$NN classifier has significantly better accuracy, kappa, and biometric performance rates for all of demographic groups when used with PCA+SOM limited data dimensions. These results guided our decision to implement this classifier for the remainder of our experiments.

Table 8.3. Comparison of Kappa Rates for Different Classifiers and PCA+SOM Grid Sizes

| **D#** | 10 × 10 | | | 20 × 20 | | | 30 × 30 | | | 40 × 40 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **ANN** | **DT** | $K$**NN** | **ANN** | **DT** | $K$**NN** | **ANN** | **DT** | $K$**NN** | **ANN** | **DT** | $K$**NN** |
| **D1** | 20.75% | 45.14% | 61.12% | 29.75% | 67.00% | 88.00% | 34.86% | 79.05% | 94.82% | 35.60% | 85.45% | 97.47% |
| **D2** | 17.32% | 41.77% | 61.95% | 27.50% | 66.50% | 87.20% | 32.65% | 77.73% | 94.90% | 33.49% | 84.62% | 97.47% |
| **D3** | 21.72% | 42.96% | 61.86% | 31.79% | 68.47% | 87.96% | 35.30% | 78.99% | 94.85% | 36.88% | 84.78% | 97.45% |
| **D4** | 12.25% | 47.75% | 37.34% | 16.67% | 69.64% | 71.71% | 17.72% | 80.34% | 84.41% | 19.79% | 86.21% | 90.77% |
| **D5** | 15.19% | 48.44% | 60.50% | 21.48% | 70.19% | 85.05% | 25.82% | 80.57% | 92.87% | 26.17% | 86.28% | 96.21% |
| **D6** | 27.49% | 52.71% | 66.59% | 44.58% | 74.33% | 88.61% | 50.23% | 83.27% | 95.01% | 51.48% | 87.90% | 97.43% |
| **D7** | 20.02% | 41.40% | 63.93% | 28.62% | 67.58% | 88.07% | 32.75% | 79.24% | 94.71% | 33.51% | 84.73% | 97.37% |
| **D8** | 13.09% | 46.41% | 63.33% | 22.45% | 68.07% | 87.47% | 26.40% | 79.07% | 94.81% | 29.74% | 85.47% | 97.32% |
| **D9** | 16.47% | 44.60% | 60.36% | 23.29% | 67.99% | 86.35% | 26.50% | 79.25% | 93.72% | 28.27% | 85.32% | 96.77% |
| **D10** | 14.72% | 44.89% | 60.14% | 20.79% | 68.62% | 86.16% | 23.63% | 79.26% | 93.81% | 24.11% | 85.05% | 96.77% |
| **D11** | 9.81% | 44.18% | 58.39% | 15.81% | 67.66% | 85.24% | 18.67% | 78.52% | 93.23% | 18.83% | 84.97% | 96.44% |

## 8.4. Different Number of Users

In order to gain more insight into the impact of the number of users on our approach performance, we observed how the kappa of demographic predictions are influenced by increasing the number of subjects (Kwapisz et al., 2010). We used the kappa metric because it is a more useful measure for problems that have an imbalance of data. This experiment selected demographic groups (D1, D2, D3, D6, and D7) to record the kappa rates for each different number of user data sets. We began the experiment with ten subjects in order to have enough data samples to validate. Then, we increased the number of subjects by ten until we reached the total number of available subject data samples. For each increasing number of subjects, we performed cross-validation to produce the kappa rates as it shown in Table 8.5. This Table shows that the kappa rates decrease gradually when we add more users.

## 8.5. Different Number of $K$-Nearest Neighbors

The $K$NN classifier performance is usually sensitive to its $K$ nearest neighbors. The $K$ neighbors of the previous experiments depended on set the $K$ neighbors equal to the number

Table 8.4. Comparison of Biometric Performance Rates for Different Classifiers

| **D#** | **Sensitivity** | | | **Specificity** | | | **ROC** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **ANN** | **DT** | ***K*NN** | **ANN** | **DT** | ***K*NN** | **ANN** | **DT** | ***K*NN** |
| **D1** | 41.30% | 89.50% | 98.36% | 90.69% | 95.66% | 99.14% | 77.69% | 93.35% | 99.43% |
| **D2** | 33.76% | 87.66% | 98.10% | 93.81% | 96.31% | 99.33% | 77.13% | 92.92% | 99.40% |
| **D3** | 54.82% | 90.82% | 98.48% | 81.43% | 93.86% | 98.93% | 76.00% | 93.12% | 99.42% |
| **D4** | 63.44% | 91.12% | 98.74% | 89.30% | 95.35% | 99.32% | 86.08% | 93.85% | 99.59% |
| **D5** | 98.43% | 98.81% | 99.67% | 39.65% | 86.11% | 98.00% | 87.03% | 94.21% | 99.41% |
| **D6** | 43.88% | 87.20% | 98.00% | 98.17% | 98.88% | 99.65% | 88.89% | 94.37% | 99.40% |
| **D7** | 66.22% | 91.99% | 98.69% | 74.29% | 93.00% | 98.76% | 77.72% | 93.20% | 99.42% |
| **D8** | 38.24% | 86.98% | 98.00% | 94.84% | 96.75% | 99.34% | 79.81% | 93.31% | 99.39% |
| **D9** | 65.58% | 92.23% | 98.87% | 83.61% | 94.37% | 99.08% | 83.49% | 93.94% | 99.56% |
| **D10** | 54.99% | 90.99% | 98.61% | 86.74% | 94.83% | 99.05% | 80.19% | 93.59% | 99.48% |
| **D11** | 62.71% | 91.71% | 98.75% | 81.74% | 94.37% | 99.10% | 80.26% | 93.87% | 99.53% |

of the classification problems of Table 1.1. In this experiment, we selected five demographic groups (D1, D5, D8, D9, and D11) with different $K$ neighbors values ranging from one to ten. Table 8.6 shows the accuracy rates of the ten $K$ neighbors for each demographic group. The accuracy rates decrease somewhat gradually when we add more $K$ neighbors. The best accuracy rates were achieved when the $K$ neighbor value is one. Based on this result, if better accuracy rates are desired, we would generalize our selection of the $K$ neighbors to be equal to one because each of the training vectors defines a region in the space.

Table 8.5. Kappa Rates Versus Number of Subjects

| **# of subjects** | **D1** | **D2** | **D3** | **D6** | **D7** |
|---|---|---|---|---|---|
| **10** | 99.23% | 98.85% | 99.27% | 99.06% | 98.92% |
| **20** | 98.61% | 98.60% | 98.71% | 98.74% | 98.64% |
| **30** | 98.28% | 98.22% | 98.41% | 98.39% | 98.29% |
| **40** | 98.12% | 98.13% | 98.13% | 98.12% | 98.00% |
| **50** | 97.89% | 97.92% | 97.92% | 97.96% | 97.88% |
| **60** | 97.77% | 97.80% | 97.74% | 97.79% | 97.72% |
| **70** | 97.69% | 97.72% | 97.66% | 97.73% | 97.66% |
| **80** | 97.57% | 97.63% | 97.53% | 97.61% | 97.55% |
| **90** | 97.53% | 97.52% | 97.45% | 97.51% | 97.38% |
| **100** | 97.47% | 97.47% | 97.45% | 97.43% | 97.37% |

Table 8.6. Accuracy Rates Versus Number of $K$ Neighbors

| **# of $K$** | **D1** | **D5** | **D8** | **D9** | **D11** |
|---|---|---|---|---|---|
| **1** | 99.22% | 99.24% | 99.19% | 98.70% | 98.58% |
| **2** | 98.89% | 98.88% | 98.85% | 98.14% | 97.97% |
| **3** | 98.86% | 98.87% | 98.82% | 98.06% | 97.86% |
| **4** | 98.63% | 98.67% | 98.58% | 97.73% | 97.49% |
| **5** | 98.55% | 98.58% | 98.49% | 97.53% | 97.27% |
| **6** | 98.36% | 98.42% | 98.32% | 97.26% | 96.98% |
| **7** | 98.27% | 98.32% | 98.20% | 97.07% | 96.74% |
| **8** | 98.09% | 98.17% | 98.04% | 96.83% | 96.49% |
| **9** | 97.99% | 98.05% | 97.93% | 96.64% | 96.27% |
| **10** | 97.84% | 97.92% | 97.78% | 96.42% | 96.04% |

# Chapter 9
# DISCUSSION

This chapter highlights the importance of the demographic group classification approach.. The chapter discusses the limitations and extensions for our proposed approach. Later, the chapter discusses the study conclusion and future work.

## 9.1. Limitations and Potential Extensions

### 9.1.1. Influence of Using Multiple Devices

This study was based on collecting data from a single tablet device. Due to the differences that exist from one device to another, the study should consider collecting data samples from multiple smart devices in order to capture variety of data. This includes both different devices of the same model where variation among the internal sensors may be present as well as the use of different tablet models. In terms of using multiple versions of the same device, variation of accuracy rates with respect to different tolerances present in the internal sensors could be established. In terms of using different models and manufacturers of smart devices, an opportunity to compare and identify the device model in user by a subject could be studied.

### 9.1.2. Applying Different Screen Sizes on the User Device

As we carried out our study on a relatively large-screen device to aid users in viewing the application UIs more clearly. We believe that a smaller-sized screen such as those found in other mobile devices may help in capturing more user gesture data. In the case of a smaller-sized screen, users would likely move the screen content around more frequently and thus our approach would likely be enabled to record more observations over time. In contrast, for the case of a larger-sized screen, users can read or view the screen content for a longer period of time without scrolling and thus our approach would likely record fewer gesture observations over time. The larger-sized screen also introduces more degrees of freedom than the smaller-sized screen (Frank et al., 2013). Furthermore, some mobile devices might have more sensors that can be exploited for extracting user gesture data than other devices. The authors do note that the number of embedded sensors within any mobile device tends to increase as newer models are designed and offered for sale to the public.

### 9.1.3. Influence of the Application Familiarity

A limitation of our study is that it only considered collecting user gesture data when the test subject interacted with the data collection application for the first time. In one respect, this allowed us to neglect any potential biases that may be due to application familiarity. However, as a subject gains familiarity with a particular application, their corresponding gesture behavior will likely change. This biometric characteristic has been established to be present in studies where dynamic keyboard biometric applications indicate that a string entered often by a user, such as their name, develops distinctive and reproducible timing and pressure characteristics (Thornton, 2011). A study where data is captured from subjects that gain increasing familiarity with a given application would allow for characterization of how captured gesture data evolves with respect to application familiarity (Sitová et al., 2016).

### 9.1.4. Combining the Approach with Other Device Information

Another limitation of the study is that we have not considered fusing our predictive results with deterministically available data within the device. For example, if the GPS receiver is enabled, our predicted classes could be combined with the known geolocation in a post-processing phase that could refine our overall prediction.

### 9.1.5. Utilizing Additional Sensors

Our approach is based upon integrating or fusing three different commonly used device sensors. One potential extension is to utilize additional sensors such as the gyroscope, magnetometer, heart rate, and other sensors that are becoming common in modern devices. By adding more sensor data to our initial collection phase, we could extract more features into the MF data space which may increase the predication accuracy results. For example, the heart rate sensor may be useful for the age demographic because younger users usually have a higher heart rate than older users.

### 9.1.6. Categorizing Additional User Demographic Groups

Our demographic group prediction method depends upon categorizing users into groups. In our study, we categorized subject demographics based on samples from a university population and we used a predetermined set of demographic group cardinalities. Hence, the categorizing mechanism may be different in other domains where other subject populations are included. Some populations may include more subjects with lower educational levels or different percentages of native languages. In terms of set cardinality, it may be desirable to use different cardinalities associated with each demographic group. For example, it may be desired to categorize subject gender with respect to a set of cardinality three such as male, female, and others. Categorizing additional user groups necessarily results in adding additional classification problems. Our approach yielded results indicating that increasing the number of the classification problems may not be a drastically limiting constraint. This is because our approach applies the aforementioned data reduction mechanisms (i.e., PCA and SOM) to reduce the subject group data to support the classifier performance. For example, D4 resulted in twenty-eight classification problems, and the PCA+SOM techniques improved the $K$NN classifier accuracy rates to have 91.91% as shown in Table 8.2.

### 9.1.7. Applying Different Biometric Approaches

Our study was based on a behavioral-based approach as discussed previously. A potential extension could be to apply a dynamic-based approach to compare the effectiveness of this study. In order to implement the dynamics-based approach, future investigators would have to build a programmable framework to generate more statistical features as in (de Mendizabal-Vzquez et al., 2014; Frank et al., 2013; Kwapisz et al., 2010).

### 9.1.8. Improving the ML Technique

The use of ML is a core method used in our approach. There are several potential improvements for ML techniques. One improvement is to use methods similar to those used in linear quadratic estimation (LQE) (i.e., Kalman filtering) or others whereby the predictive phase of our current method is followed by a post processing phase such that subsequent predictions are updated using a weighted average or other measures from previous predictions. Another improvement could be to add $k$-means or hierarchical clustering techniques after SOM processing. Ideally, clustering techniques can be implemented on SOM nodes to isolate groups of samples with similar metrics. This might improve the accuracy rates for some of the chosen classifiers, such as the ANN classifier.

## 9.2. Conclusion

In this investigation, we describe an effective mechanism for demographic group predictions among smart device users based on user gestures and the response of the internal device sensors. An advantage of this approach is that it does not require any additional specialized hardware since it depends only on the sensors already present in most modern smart devices. Our approach has many potential applications ranging from serving customized advertisements or other data, continuous and non-obtrusive authentication, validation of explicitly entered demographic data, and others. Our ML approach integrates the data from multiple on-board sensors, pre-processes and reduces the data into a smaller-dimensioned space using PCA+SOM, and then performs supervised demographic classification. Experimental accuracy rates indicate that the best results are achieved by implementing PCA+SOM techniques for different demographic groups prior to invoking the $K$NN classification method. The achieved accuracy rates were all greater than 90% for the demographics under consideration.

### 9.3. Future Work

In future work, we plan to continue capturing more subject data samples to refine our results and to observe the accuracy rate versus the number of users. We will also continue to add more demographic groups that may be discernible using our approach. Our future work will also consider using more and different data analysis, visualization, and statistical hypothesis tests. We also plan to investigate the use of alternative classifiers to determine performance versus accuracy trade offs regarding ML approaches for demographic group predictions.

# Appendix A

# THE CAPTURE DEVICE TOOL

This chapter provides information about the capture device tool that was used in the study in Table A.1 and Figure A.1. More information can be found in (ASUS).

Table A.1. The Capture Device Tool Information

| Features | Details |
|---|---|
| **Body Weight** | 302 g (10.65 oz) |
| **Display Type** | IPS LCD capacitive touch-screen, 16M colors |
| **Display Size** | 7.0 inches ( 59.9% screen-to-body ratio) |
| **Display Resolution** | 800 x 1280 pixels ( 216 ppi pixel density) |
| **Platform OS** | Android OS, v4.2 (Jelly Bean), upgradable to v4.2.2 (Jelly Bean) |
| **Platform CPU** | Quad-core 1.2 GHz Cortex-A7 |
| **Card slot Memory** | Micro SD, up to 32 GB |
| **Internal Memory** | 16 GB, 1 GB RAM |

Figure A.1. A Picture of The ASUS MeMO Pad HD 7 (ME173X)

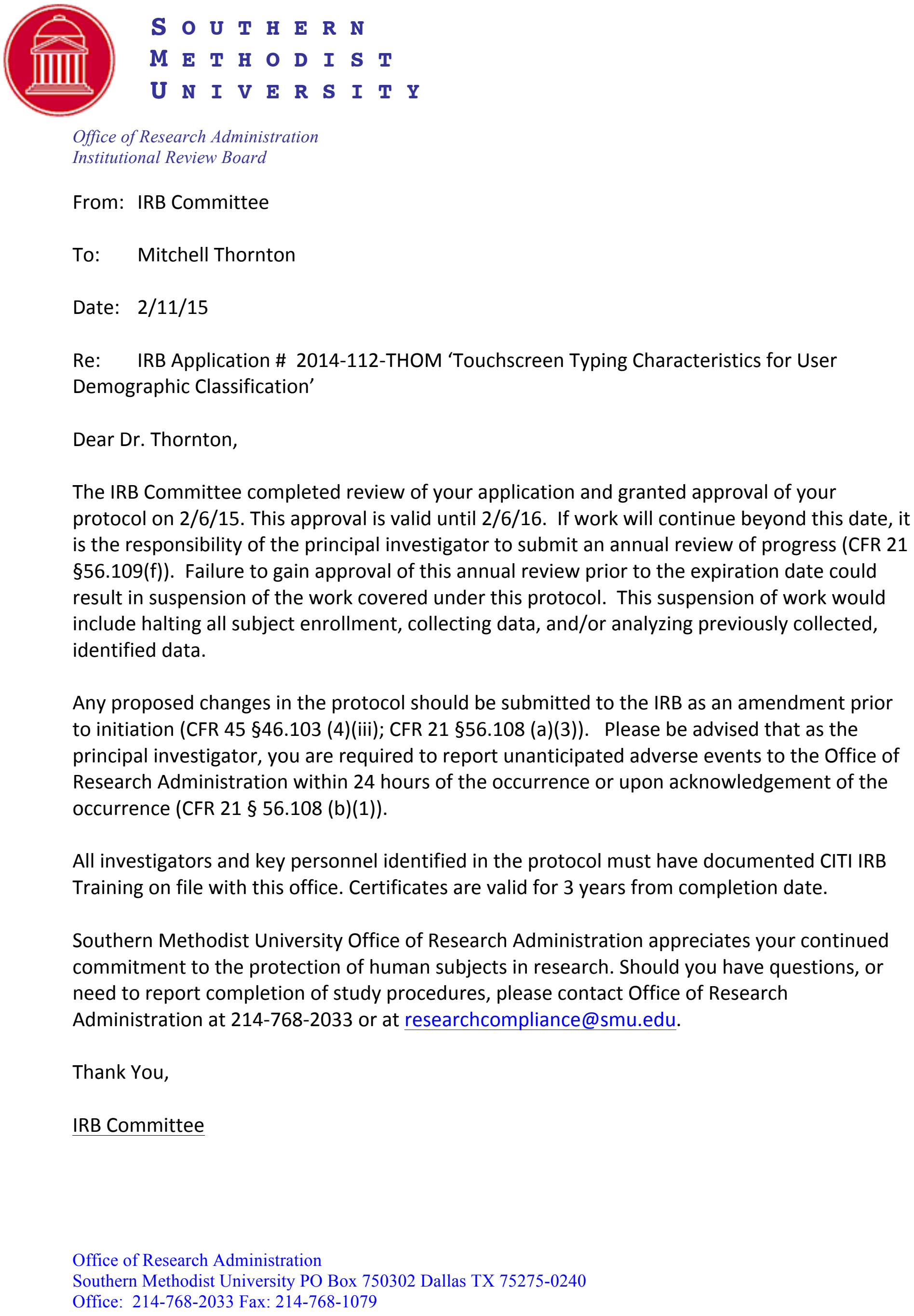
# Appendix B

## SMU RESEARCH COMPLIANCE

This chapter explains the SMURC agreement terms and provides a copy of the IRB committee approval. The agreement terms were as follows:

- **Consent Form:** Each participant must fill out this form before beginning the data collection process. It shows that volunteers are willing for us to collect their behavioral data while using our classy application. The link for the form can be found here
- **Disclosure Form:** After participants have given their behavioral data and the research has been done completely, the volunteers should receive this form by email. The disclosure form explains the research goals and the information about the research.
- **Survey Form:** Each participant must fill out this form after the data collection process. It records the volunteers' information such as name, ages, language, nationally, and more important information for the study. The link for the form can be found here

Figure B.1 shows a copy of the IRB committee approval



SOUTHERN
METHODIST
UNIVERSITY

*Office of Research Administration*
*Institutional Review Board*

From: IRB Committee

To: Mitchell Thornton

Date: 2/11/15

Re: IRB Application # 2014-112-THOM 'Touchscreen Typing Characteristics for User Demographic Classification'

Dear Dr. Thornton,

The IRB Committee completed review of your application and granted approval of your protocol on 2/6/15. This approval is valid until 2/6/16. If work will continue beyond this date, it is the responsibility of the principal investigator to submit an annual review of progress (CFR 21 §56.109(f)). Failure to gain approval of this annual review prior to the expiration date could result in suspension of the work covered under this protocol. This suspension of work would include halting all subject enrollment, collecting data, and/or analyzing previously collected, identified data.

Any proposed changes in the protocol should be submitted to the IRB as an amendment prior to initiation (CFR 45 §46.103 (4)(iii); CFR 21 §56.108 (a)(3)). Please be advised that as the principal investigator, you are required to report unanticipated adverse events to the Office of Research Administration within 24 hours of the occurrence or upon acknowledgement of the occurrence (CFR 21 § 56.108 (b)(1)).

All investigators and key personnel identified in the protocol must have documented CITI IRB Training on file with this office. Certificates are valid for 3 years from completion date.

Southern Methodist University Office of Research Administration appreciates your continued commitment to the protection of human subjects in research. Should you have questions, or need to report completion of study procedures, please contact Office of Research Administration at 214-768-2033 or at researchcompliance@smu.edu.

Thank You,

IRB Committee

Office of Research Administration
Southern Methodist University PO Box 750302 Dallas TX 75275-0240
Office: 214-768-2033 Fax: 214-768-1079

Figure B.1. A Copy of The IRB Committee Approval

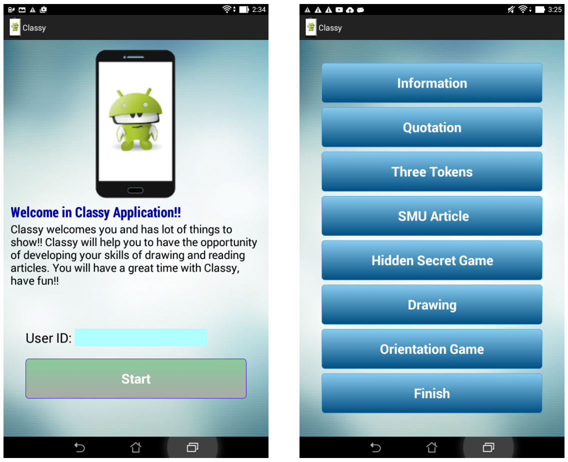
# Appendix C

# DEMOGRAPHIC APPLICATION INTERFACES

This chapter provides the demographic group application UIs as mentioned in chapter 4.

## C.1. The Start Up Interfaces

In our study, the participant asked to use demographic group application. Figure C.1 shows the first interface (main) that appears to the participant , which contains a welcome message and some information about the application. In order to start up the application the participant required to enter a User-ID number. The data collector gives the participant corresponding to the same number that is used to fill out the online forms before. After, the participant start up the application, a list of task options appears to select any task that the participant likes.



Welcome in Classy Application!!

Classy welcomes you and has lot of things to show!! Classy will help you to have the opportunity of developing your skills of drawing and reading articles. You will have a great time with Classy, have fun!!

User ID:

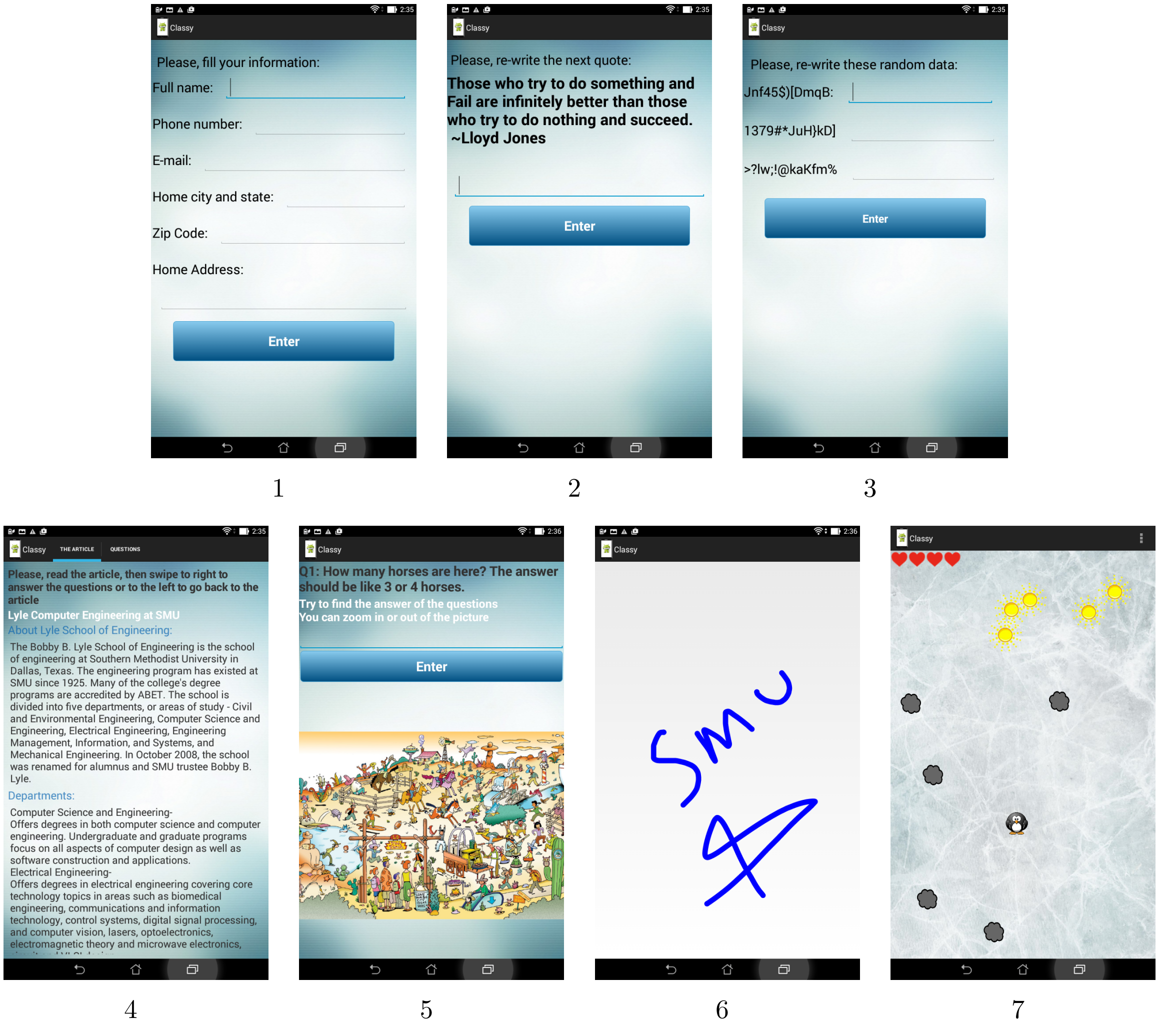
Main

Options

Figure C.1. The Start Up Interfaces

## C.2. The Task Interfaces

All of the participants in the study required to use the application seven task interfaces. Figure C.2 shows the seven UIs of our application. Each of the screen encoded with a UI number that UI number is corresponded to the UI description in Table 4.1.



Please, fill your information:

Full name:

Phone number:

E-mail:

Home city and state:

Zip Code:

Home Address:

1

Please, re-write the next quote:

**Those who try to do something and Fail are infinitely better than those who try to do nothing and succeed. ~Lloyd Jones**

2

Please, re-write these random data:

Jnf45$)[DmqB:

1379#*JuH}kD]

>?lw;!@kaKfm%

3

**Please, read the article, then swipe to right to answer the questions or to the left to go back to the article**

**Lyle Computer Engineering at SMU**

About Lyle School of Engineering:

The Bobby B. Lyle School of Engineering is the school of engineering at Southern Methodist University in Dallas, Texas. The engineering program has existed at SMU since 1925. Many of the college's degree programs are accredited by ABET. The school is divided into five departments, or areas of study - Civil and Environmental Engineering, Computer Science and Engineering, Electrical Engineering, Engineering Management, Information, and Systems, and Mechanical Engineering. In October 2008, the school was renamed for alumnus and SMU trustee Bobby B. Lyle.

Departments:

Computer Science and Engineering-
Offers degrees in both computer science and computer engineering. Undergraduate and graduate programs focus on all aspects of computer design as well as software construction and applications.
Electrical Engineering-
Offers degrees in electrical engineering covering core technology topics in areas such as biomedical engineering, communications and information technology, control systems, digital signal processing, and computer vision, lasers, optoelectronics, electromagnetic theory and microwave electronics,

4

**Q1: How many horses are here? The answer should be like 3 or 4 horses.**

**Try to find the answer of the questions**
**You can zoom in or out of the picture**

5

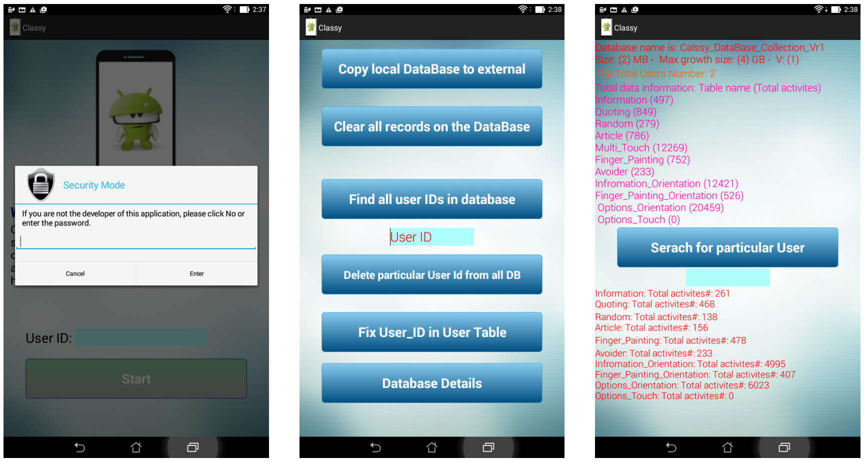
6

7

Figure C.2. The Seven Task Interfaces

## C.3. The Additional Interfaces

The application have additional characteristics that allows the data collector to have more control and management as explained in 4.3. Figure C.3 shows the security method that is used where the data collector accesses the control options by entering password. Then, the data collector can copy or remove the data from the application to the device's SD card. It also allows the data collector to check the collected data by entering the participant's User-ID number.


Classy
Security Mode
Cancel
Enter
User ID:
Start
Classy
Copy local DataBase to external
Clear all records on the DataBase
Find all user IDs in database
User ID
Delete particular User Id from all DB
Fix User_ID in User Table
Database Details
Classy
Serach for particular User

If you are not the developer of this application, please click No or enter the password.

```
Database name is: Calssy_DataBase_Collection_Vr1
Size: (2) MB -  Max growth size: (4) GB -  V: (1)
The Total Users Number: 2
Total data information: Table name (Total activites)
Information (497)
Quoting (849)
Random (279)
Article (786)
Multi_Touch (12269)
Finger_Painting (752)
Avoider (233)
Infromation_Orientation (12421)
Finger_Painting_Orientation (526)
 Options_Orientation (20459)
 Options_Touch (0)

Information: Total activites#: 261
Quoting: Total activites#: 468
Random: Total activites#: 138
Article: Total activites#: 156
Finger_Painting: Total activites#: 478
Avoider: Total activites#: 233
Infromation_Orientation: Total activites#: 4995
Finger_Painting_Orientation: Total activites#: 407
Options_Orientation: Total activites#: 6023
Options_Touch: Total activites#: 0
```

Security    Control    Management

Figure C.3. The Additional Interfaces

## Appendix D

## THE METHODOLOGY *R* SOURCE CODE

This chapter provides the *R* code is used through the study's methodology. This code required a fundamental understanding of the *R* language and the *R* studio software tool. The chapter is divided based on the study's methodology implementing sections. Each section should be implemented separately.

### D.1. Data Integration

```
# Connect R to sqlite
library(sqldf)
library(XLConnect)
# Choose the database file path to connect into
dbfile = "E:\\demo\\Calssy_DataBase_Collection_Vr8"
# Instantiate the db Driver to a convenient object
sqlite = SQLite()
# Assign the connection string to a connection object
db <- dbConnect(sqlite, dbfile)
# Select user_id
user_id = 77

# Sqlite commands
# Key-Strokes
KS.UI.1 <- paste("SELECT Key_Action, Key_Code, Time
                  FROM Information WHERE User_ID =",user_id, sep="");
KS.UI.2 <- paste("SELECT Key_Action, Key_Code, Time
                  FROM Quoting WHERE User_ID =",user_id, sep="");
```

```
KS.UI.3 <- paste("SELECT Key_Action, Key_Code, Time
                 FROM Random WHERE User_ID =",user_id, sep="");
# Touch-Screen (Single finger)
TS.UI.1 <- paste("SELECT Actions, Xcoordination, Ycoordination,
                  Pressure, Size, Time FROM Article
                 WHERE User_ID =",user_id, sep="");
TS.UI.2 <- paste("SELECT Actions, Xcoordination, Ycoordination,
                  Pressure, Size, Time FROM Finger_Painting
                 WHERE User_ID =",user_id, sep="");
# Touch-Screen (Multi-fingers)
TS.UI.3 <- paste("SELECT Actions, Xcoordination0, Ycoordination0,
                  Xcoordination1, Ycoordination1, Pressure0, Pressure1,
                  Size0, Size1, Time FROM Multi_Touch
                 WHERE User_ID =",user_id, sep="");
# Accelerometer
AC.UI.1 <- paste("SELECT X, Y, Z, Time FROM Avoider
                 WHERE User_ID =",user_id, sep="");
AC.UI.2 <- paste("SELECT X, Y, Z, Time FROM Finger_Painting_Orientation
                 WHERE User_ID =",user_id, sep="");
AC.UI.3 <- paste("SELECT X, Y, Z, Time FROM Infromation_Orientation
                 WHERE User_ID =",user_id, sep="");


# Data frames
# Key-Strokes
Data.KS.UI.1 <- dbGetQuery(db, KS.UI.1)
Data.KS.UI.2 <- dbGetQuery(db, KS.UI.2)
Data.KS.UI.3 <- dbGetQuery(db, KS.UI.3)
# Touch-Screen (Single-finger)
Data.TS.UI.1 <- dbGetQuery(db, TS.UI.1)
Data.TS.UI.2 <- dbGetQuery(db, TS.UI.2)
# Touch-Screen (Multi-fingers)
Data.TS.UI.3 <- dbGetQuery(db, TS.UI.3)
```

```
# Accelerometer
Data.AC.UI.1 <- dbGetQuery(db, AC.UI.1)
Data.AC.UI.2 <- dbGetQuery(db, AC.UI.2)
Data.AC.UI.3 <- dbGetQuery(db, AC.UI.3)

# Combined all the data frames
library(qpcR)
combine <- qpcR:::cbind.na(Data.KS.UI.1,Data.KS.UI.2, Data.KS.UI.3,
                           Data.TS.UI.1, Data.TS.UI.2, Data.TS.UI.3,
                           Data.AC.UI.1, Data.AC.UI.2, Data.AC.UI.3)
# Insert user_id into the combined data frame
combine$User_ID = user_id

# Rename the colname names
colnames(combine) <- c(”F1”,”F2”,”F3”,
                       ”F4”,”F5”,”F6”,
                       ”F7”,”F8”,”F9”,
                       ”F10”,”F11”,”F12”,”F13”,”F14”,”F15”,
                       ”F16”,”F17”,”F18”,”F19”,”F20”,”F21”,
                       ”F22”,”F23”,”F24”,”F25”,”F26”,”F27”,
                       ”F28”,”F29”,”F30”,”F31”,
                       ”F32”,”F33”,”F34”,”F35”,
                       ”F36”,”F37”,”F38”,”F39”,
                       ”F40”,”F41”,”F42”,”F43”, ”User_ID”)

# Saving the user combined data frame into the Integration file
user.name <- paste(”User”, user_id, ”.csv”, sep = ””)
user.file <- file.path(”E:\\demo\\Integration”, user.name)
write.csv(combine, user.file , row.names= FALSE)
```

### D.2. Data Pre-processing

```
# Select the user ID number
user_id = 77
# Find the user file from the Integration file
user.file <- paste(”User”, user_id, sep = ””)
path <- paste(”E:\\demo\\Integration\\”, user.file, ”.csv”, sep = ””)
# Load the data
Data <- read.csv(path, sep=”,”, quote=’”’, header=T, row.names=NULL)
# Prepare the user data by dividing the features into data frames
# Key-Stroks
KS.UI.1 <- Data[1:length(na.omit(Data[,3])), 1:3]
KS.UI.2 <- Data[1:length(na.omit(Data[,6])), 4:6]
KS.UI.3 <- Data[1:length(na.omit(Data[,9])), 7:9]
# Touch-Screen (Single-finger)
TS.UI.1 <- Data[1:length(na.omit(Data[,15])), 10:15]
TS.UI.2 <- Data[1:length(na.omit(Data[,21])), 16:21]
# Touch-Screen (Multi-fingers)
TS.UI.3 <- Data[1:length(na.omit(Data[,31])), 22:31]
# Accelerometer
AC.UI.1 <- Data[1:length(na.omit(Data[,35])), 32:35]
AC.UI.2 <- Data[1:length(na.omit(Data[,39])), 36:39]
AC.UI.3 <- Data[1:length(na.omit(Data[,43])), 40:43]

# Pre-processing 1
# Change to numeric
# Key-Stroks
KS.UI.1[,1] <- as.numeric(KS.UI.1[,1])
KS.UI.2[,1] <- as.numeric(KS.UI.2[,1])
KS.UI.3[,1] <- as.numeric(KS.UI.3[,1])
# Touch-Screen (Single-finger)
TS.UI.1[,1] <- as.numeric(TS.UI.1[,1])
TS.UI.2[,1] <- as.numeric(TS.UI.2[,1])
```

```
# Touch-Screen (Multi-fingers)
TS.UI.3[,1] <- as.numeric(TS.UI.3[,1])

# Pre-processing 2
# Repalce missing values only on (Multi-fingers)
TS.UI.3[is.na(TS.UI.3)] <- 0

# Pre-processing 3
# Create the constant value function
mean_removal <- function (sensor, sensor_mean){
  new.sensor <- NULL
  for(i in 1:length(sensor)){
    new.sensor[i] <- sensor[i] - sensor_mean }
  new.sensor
}
# Apply the mean_removal function
AC.UI.1[,1] <- mean_removal(AC.UI.1[,1], mean(AC.UI.1[,1]))
AC.UI.1[,2] <- mean_removal(AC.UI.1[,2], mean(AC.UI.1[,2]))
AC.UI.1[,3] <- mean_removal(AC.UI.1[,3], mean(AC.UI.1[,3]))
AC.UI.2[,1] <- mean_removal(AC.UI.2[,1], mean(AC.UI.2[,1]))
AC.UI.2[,2] <- mean_removal(AC.UI.2[,2], mean(AC.UI.2[,2]))
AC.UI.2[,3] <- mean_removal(AC.UI.2[,3], mean(AC.UI.2[,3]))
AC.UI.3[,1] <- mean_removal(AC.UI.3[,1], mean(AC.UI.3[,1]))
AC.UI.3[,2] <- mean_removal(AC.UI.3[,2], mean(AC.UI.3[,2]))
AC.UI.3[,3] <- mean_removal(AC.UI.3[,3], mean(AC.UI.3[,3]))

# Pre-processing 4
# Smoothing with the weighted moving average (WMA)
library(TTR)
AC.UI.1[,1] <- WMA(AC.UI.1[,1], n=5, wts=1:length(AC.UI.1[,1]))
AC.UI.1[,2] <- WMA(AC.UI.1[,2], n=5, wts=1:length(AC.UI.1[,2]))
AC.UI.1[,3] <- WMA(AC.UI.1[,3], n=5, wts=1:length(AC.UI.1[,3]))
AC.UI.2[,1] <- WMA(AC.UI.2[,1], n=5, wts=1:length(AC.UI.2[,1]))
AC.UI.2[,2] <- WMA(AC.UI.2[,2], n=5, wts=1:length(AC.UI.2[,2]))
AC.UI.2[,3] <- WMA(AC.UI.2[,3], n=5, wts=1:length(AC.UI.2[,3]))
AC.UI.3[,1] <- WMA(AC.UI.3[,1], n=5, wts=1:length(AC.UI.3[,1]))
AC.UI.3[,2] <- WMA(AC.UI.3[,2], n=5, wts=1:length(AC.UI.3[,2]))
AC.UI.3[,3] <- WMA(AC.UI.3[,3], n=5, wts=1:length(AC.UI.3[,3]))
# Remove the noise values
AC.UI.1 <- na.omit (AC.UI.1)
AC.UI.2 <- na.omit (AC.UI.2)
AC.UI.3 <- na.omit (AC.UI.3)

# Pre-processing 5
# Normalization
for (r in 2:3) {KS.UI.1[,r] <- scale(KS.UI.1[,r])}
for (r in 2:3) {KS.UI.2[,r] <- scale(KS.UI.2[,r])}
for (r in 2:3) {KS.UI.3[,r] <- scale(KS.UI.3[,r])}
for (r in 2:6) {TS.UI.1[,r] <- scale(TS.UI.1[,r])}
for (r in 2:6) {TS.UI.2[,r] <- scale(TS.UI.2[,r])}
for (r in 2:10) {TS.UI.3[,r] <- scale(TS.UI.3[,r])}
for (r in 1:4) {AC.UI.1[,r] <- scale(AC.UI.1[,r])}
for (r in 1:4) {AC.UI.2[,r] <- scale(AC.UI.2[,r])}
for (r in 1:4) {AC.UI.3[,r] <- scale(AC.UI.3[,r])}

# Pre-processing 6
# Scale or range data values from zero to one
range01 <- function(x, na.rm = FALSE){
  (x-min(x))/abs(max(x)-min(x)) }
# Apply the range01 fuction
for (r in 2:3) {KS.UI.1[,r] <- range01(KS.UI.1[,r])}
for (r in 2:3) {KS.UI.2[,r] <- range01(KS.UI.2[,r])}
for (r in 2:3) {KS.UI.3[,r] <- range01(KS.UI.3[,r])}
for (r in 2:6) {TS.UI.1[,r] <- range01(TS.UI.1[,r])}
for (r in 2:6) {TS.UI.2[,r] <- range01(TS.UI.2[,r])}
for (r in 2:10) {TS.UI.3[,r] <- range01(TS.UI.3[,r])}
for (r in 1:4) {AC.UI.1[,r] <- range01(AC.UI.1[,r])}
for (r in 1:4) {AC.UI.2[,r] <- range01(AC.UI.2[,r])}
for (r in 1:4) {AC.UI.3[,r] <- range01(AC.UI.3[,r])}

# Pre−processing 7
# Equalization
library(qpcR)
Features <- qpcR:::cbind.na(KS.UI.1, KS.UI.2, KS.UI.3,
                            TS.UI.1, TS.UI.2, TS.UI.3,
                            AC.UI.1, AC.UI.2, AC.UI.3)
# Delete row.names
row.names(Features) <- NULL
Features <- as.data.frame (Features)
# Replace features with zero to equlized the number of sample sizes
Features[is.na(Features)] <- 0
# Add the user_id into the feature
Features$User_ID <- Data[1:nrow(Features), 44]

# Pre−processing 8
# Labeling
lookup <- read.csv(”E:\\demo\\Demographics.csv”, sep=”,”, quote='”', header=T)
library(plyr)
user.demographic <- join(Features, lookup, by = ”User_ID”)

# Saving the user file demographic data frame into the Preprocessing file
user.name <- paste(”User”, user_id, ”.csv”, sep = ””)
user.file <- file.path(”E:\\demo\\Preprocessing”, user.name)
write.csv(user.demographic, user.file , row.names= FALSE)
```

## D.3. Data Reduction

This section is divided into two parts as:

### D.3.1. PCA

```
# Select the user ID number
user_id = 77
# Find the user from the Preprocessing file
user.file <- paste("User", user_id, sep = "")
path <- paste("E:\\demo\\Preprocessing\\", user.file , ".csv", sep = "")
# Load the data
Data.load <- read.csv(path, sep=",", quote='"', header=T, row.names=NULL)
# Select the features
Data <- Data.load[,1:43]

# Perform PCA
pca <- princomp(Data)
# Take the high varince components
pca.scores <- pca$scores [,1:15]

# Create user PCA data frame
user.pca <- data.frame (pca.scores, Data.load[,44:57])

# Saving the user PCA data frame into the Reduction file
user.name <- paste("User", user_id, ".csv", sep = "")
user.file <- file.path("E:\\demo\\Reduction\\PCA", user.name)
write.csv(user.pca, user.file , row.names= FALSE)
```

D.3.2. SOM

```
# Select the user ID number
user_id = 77
# Find the user from the PCA file
user.file <- paste("User", user_id, sep = "")
path <- paste("E:\\demo\\Reduction\\PCA\\", user.file , ".csv", sep = "")
# Load the data
Data.load <- read.csv(path, sep=",", quote='"', header=T, row.names=NULL)
# Select the features and convert it to be a matrix
Data <- as.matrix(Data.load[,1:15])

# Perform SOM
library(kohonen)
# Set up the seed
set.seed(400)

# Set up the SOM grid
som_grid <- somgrid(xdim = 20, ydim= 20 , topo="hexagonal")
som_model <- som(Data, grid=som_grid, rlen=500,
                 alpha=c(0.05,0.01), keep.data = TRUE, n.hood="circular" )

# Create user SOM data frame
user.som <- data.frame (som_model$codes,
                        Data.load[1:nrow(som_model$codes), 16:29])

# Saving the user SOM data frame into the Reduction file
user.name <- paste("User", user_id, ".csv", sep = "")
user.file <- file.path("E:\\demo\\Reduction\\SOM", user.name)
write.csv(user.som, user.file , row.names= FALSE)
```

### D.4. Supervised Classification

```
# Prepare the users data files into one set
# Select the path of the folder that holds multiple .csv files
folder <- "E:\\demo\\Reduction\\SOM\\"
# Create a list of all .csv files in folder
file_list <- list.files(path= folder, pattern= "*.csv")
# Read in each .csv file in file_list and rbind them into a data frame
Data <- do.call("rbind",lapply(file_list, function(x)
  read.csv(paste(folder, x, sep=''), stringsAsFactors = FALSE)))

# Prepare data for validation
# Select the features
Data.sel <- Data[,1:15]
# Create three demographic data frames
D1 <- data.frame (Data.sel , Class = Data [,19])
D2 <- data.frame (Data.sel , Class = Data [,17])
D3 <- data.frame (Data.sel , Class = Data [,23])

# Perform validation using the caret package
library(caret)
# Set up the seed
set.seed(400)
# Prepare training scheme to 10-fold-cross-validation repeated 10 times
control <- trainControl(method="repeatedcv", number= 10, repeats= 10)

# ANN models
# Set up the nnet classifier grid
my.grid <- expand.grid(.decay = 1e-05, .size = 15)
nnet.D1 <- train(Class~., data=D1, method="nnet", trControl=control,
                   tuneLength = 10, tuneGrid = my.grid, maxit=500 , trace = F)
nnet.D2 <- train(Class~., data=D2, method="nnet", trControl=control,
                   tuneLength = 10, tuneGrid = my.grid, maxit=500 , trace = F)
```

```
nnet.D3 <- train(Class~., data=D3, method=”nnet”, trControl=control,
                  tuneLength = 10, tuneGrid = my.grid, maxit=500 , trace = F)

# DT models
dt.D1 <- train(Class~., data=D1, method=”J48”, tuneLength = 10,
                trControl=control)
dt.D2 <- train(Class~., data=D2, method=”J48”, tuneLength = 10,
                trControl=control)
dt.D3 <- train(Class~., data=D3, method=”J48”, tuneLength = 10,
                trControl=control)

# KNN models
knn.D1 <- train(Class~., data=D1, method=”knn”, trControl=control,
                 tuneLength = 10, tuneGrid=data.frame(k=2))
knn.D2 <- train(Class~., data=D2, method=”knn”, trControl=control,
                 tuneLength = 10, tuneGrid=data.frame(k=2))
knn.D3 <- train(Class~., data=D3, method=”knn”, trControl=control,
                 tuneLength = 10, tuneGrid=data.frame(k=2))
```

# REFERENCES

Ahmed Awad E Ahmed and Issa Traore. A new biometric technology based on mouse dynamics. *IEEE Transactions on dependable and secure computing*, 4(3):165, 2007.

Adel R Alharbi and Mitchell A Thornton. Demographic group classification of smart device users. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 481–486. IEEE, 2015.

Adel R Alharbi and Mitchell A Thornton. Demographic group prediction based on smart device user recognition gestures. submitted, 2016.

Jeffrey David Allen, John Joseph Howard, and Mitchell Aaron Thornton. Method for subject classification using a pattern recognition input device, October 22 2011. US Patent App. 13/279,279.

Ken Arnold, James Gosling, David Holmes, and David Holmes. *The Java programming language*, volume 2. Addison-wesley Reading, 1996.

ASUS. Asus memo pad 7 (me176c). URL http://www.asus.com/Tablets/ASUS_MeMO_Pad_7_ME176C. (Date last accessed 19-March-2016).

Cynthia L Berryman-Fink and James R Wilcox. A multivariate investigation of perceptual attributions concerning gender appropriateness in language. *Sex Roles*, 9(6):663–681, 1983.

Debnath Bhattacharyya, Rahul Ranjan, Farkhod Alisherov, and Minkyu Choi. Biometric authentication: A review. *International Journal of u-and e-Service, Science and Technology*, 2(3):13–28, 2009.

Douglas Biber, Susan Conrad, and Randi Reppen. *Corpus linguistics: Investigating language structure and use*. Cambridge University Press, 1998.

Tiejun Cheng, Yanli Wang, and Stephen H Bryant. Fselector: a ruby gem for feature selection. *Bioinformatics*, 28(21):2851–2852, 2012.

I. de Mendizabal-Vzquez, D. de Santos-Sierra, J. Guerra-Casanova, and C. Snchez-vila. Supervised classification methods applied to keystroke dynamics through mobile devices. In *Security Technology (ICCST), 2014 International Carnahan Conference on*, pages 1–6, Oct 2014. doi: 10.1109/CCST.2014.6987033.

Janez Demšar, Blaž Zupan, Gregor Leban, and Tomaz Curk. *Orange: From experimental machine learning to interactive data mining*. Springer, 2004.

Hiroshi Dozono and Masanori Nakakuni. An integration method of multi-modal biometrics using supervised pareto learning self organizing maps. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 602–606. IEEE, 2008.

Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

Imola K Fodor. A survey of dimension reduction techniques, 2002.

Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *Information Forensics and Security, IEEE Transactions on*, 8(1):136–148, 2013.

Anthony J Grenga. Android based behavioral biometric authentication via multi-modal fusion. Technical report, DTIC Document, 2014.

Shyam M Guthikonda. Kohonen self-organizing maps. *Wittenberg University*, 2005.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

Jian Hu, Hua-Jun Zeng, Hua Li, Cheng Niu, and Zheng Chen. Demographic prediction based on user's browsing behavior. In *Proceedings of the 16th international conference on World Wide Web*, pages 151–160. ACM, 2007.

Shrijit S Joshi and Vir V Phoha. Competition between som clusters to model user authentication system in computer networks. In *2007 2nd International Conference on Communication Systems Software and Middleware*, pages 1–8. IEEE, 2007.

Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982.

Max Kuhn. Caret package. *Journal of Statistical Software*, 28(5), 2008.

Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Cell phone-based biometric identification. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pages 1–7. IEEE, 2010.

Nicholas D Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T Campbell. A survey of mobile phone sensing. *Communications Magazine, IEEE*, 48(9):140–150, 2010.

Chien-Cheng Lin, Chin-Chun Chang, and Deron Liang. A novel non-intrusive user authentication method based on touchscreen of smartphones. In *Biometrics and Security Technologies (ISBAST), 2013 International Symposium on*, pages 212–216. IEEE, 2013.

Roy Maxion, Kevin S Killourhy, et al. Keystroke biometrics with number-pad input. In *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*, pages 201–210. IEEE, 2010.

Reto Meier. Professional android 4 application development, 2012.

Muhammad Muaaz and Claudia Nickel. Influence of different walking speeds and surfaces on accelerometer-based biometric gait recognition. In *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on*, pages 508–512. IEEE, 2012.

Todd Neideen and Karen Brasel. Understanding statistical tests. *Journal of surgical education*, 64(2):93–96, 2007.

K. W. Nixon, Xiang Chen, Zhi-Hong Mao, Yiran Chen, and Kang Li. Mobile user classification and authorization based on gesture usage recognition. In *Design Automation Conference (ASP-DAC), 2013 18th Asia and South Pacific*, pages 384–389, Jan 2013a. doi: 10.1109/ASPDAC.2013.6509626.

Kent W Nixon, Xiang Chen, Zhi-Hong Mao, Yiran Chen, and Kang Li. Mobile user classification and authorization based on gesture usage recognition. In *Design Automation Conference (ASP-DAC), 2013 18th Asia and South Pacific*, pages 384–389. IEEE, 2013b.

Mike Owens and Grant Allen. *SQLite*. Springer, 2010.

Nazila Panahi, Mahrokh G Shayesteh, Sara Mihandoost, and Behrooz Zali Varghahan. Recognition of different datasets using pca, lda, and various classifiers. In *Application of Information and Communication Technologies (AICT), 2011 5th International Conference on*, pages 1–5. IEEE, 2011.

Mittal C Patel, Mahesh Panchal, and Himani P Bhavsar. Decorate ensemble of artificial neural networks with high diversity for classification. 2013.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. URL http://www.R-project.org. ISBN 3-900051-07-0.

Gunnar Rätsch. A brief introduction into machine learning. In *21st Chaos Communication Congress*, 2004.

Nornadiah Mohd Razali, Yap Bee Wah, et al. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *Journal of statistical modeling and analytics*, 2(1):21–33, 2011.

Leandro A Silva and Emilio Del-Moral-Hernandez. A som combined with knn for classification task. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2368–2373. IEEE, 2011.

Jennifer A Simkins-Bullock and Beth G Wildman. An investigation into the relationships between gender and language. *Sex Roles*, 24(3-4):149–160, 1991.

Sukree Sinthupinyo, Warut Roadrungwasinkul, and Charoon Chantan. User recognition via keystroke latencies using som and backpropagation neural network. In *ICCAS-SICE, 2009*, pages 3160–3165. IEEE, 2009.

Zdeňka Sitová, Jaroslav Šeděnka, Qing Yang, Ge Peng, Gang Zhou, Paolo Gasti, and Kiran S Balagani. Hmog: New behavioral biometric features for continuous authentication of smartphone users. *IEEE Transactions on Information Forensics and Security*, 11(5):877–892, 2016.

Mitchell A Thornton. Keyboard dynamics. In *Encyclopedia of Cryptography and Security*, pages 688–691. Springer, 2011.

John R Vacca. *Biometric technologies and verification systems.* Butterworth-Heinemann, 2007.

Ron Wehrens, Lutgarde MC Buydens, et al. Self-and super-organizing maps in r: the kohonen package. *J Stat Softw*, 21(5):1–19, 2007.

Graham J Williams. Rattle: a data mining gui for r. *The R Journal*, 1(2):45–55, 2009.

Jiunn-Lin Wu and I-Jing Li. The improved som-based dimensionality reducton method for knn classifier using weighted euclidean metric. *International Journal of Computer, Consumer and Control (IJ3C)*, 3(1), 2014.

Hujun Yin. The self-organizing maps: background, theories, extensions and applications. In *Computational intelligence: A compendium*, pages 715–762. Springer, 2008.

Josh Jia-Ching Ying, Yao-Jen Chang, Chi-Min Huang, and Vincent S Tseng. Demographic prediction based on users mobile behaviors. *Mobile Data Challenge*, 2012.