TOWARDS A CONTINUOUS USER AUTHENTICATION USING HAPTIC INFORMATION

by

Fawaz Abdulaziz A Alsulaiman

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the Ph.D. degree in
Computer Science

School of Electrical Engineering and Computer Science (EECS)
Faculty of Engineering
University of Ottawa

Abstract

With the advancement in multimedia systems and the increased interest in haptics to be used in interpersonal communication systems, where users can see, show, hear, tell, touch and be touched, mouse and keyboard are no longer dominant input devices. Touch, speech and vision will soon be the main methods of human computer interaction. Moreover, as interpersonal communication usage increases, the need for securing user authentication grows. In this research, we examine a user's identification and verification based on haptic information. We divide our research into three main steps. The first step is to examine a pre-defined task, namely a handwritten signature with haptic information. The user target in this task is to mimic the legitimate signature in order to be verified. As a second step, we consider the user's identification and verification based on user drawings. The user target is predefined, however there are no restrictions imposed on the order or on the level of details required for the drawing. Lastly, we examine the feasibility and possibility of distinguishing users based on their haptic interaction through an interpersonal communication system. In this third step, there are no restrictions on user movements, however a free movement to touch the remote party is expected. In order to achieve our goal, many classification and feature reduction techniques have been discovered and some new ones were proposed. Moreover, in this work we utilize evolutionary computing in user verification and identification. Analysis of haptic features and their significance on distinguishing users is hence examined.

The results show a utilization of visual features by Genetic Programming (GP) towards identity verification, with a probability equal to 50% while the remaining haptic features were utilized with a probability of approximately 50%. Moreover, with a handwritten signature application, a verification success rate of 97.93% with False Acceptance Rate (FAR) of 1.28% and 11.54% False Rejection Rate (FRR) is achieved with the utilization of genetic programming enhanced with the random over sampled data set. In addition, with a totally free user movement in a haptic-enabled interpersonal communication system, an identification success rate of 83.3% is achieved when random forest classifier is utilized.

Acknowledgements

Thanks to Allah for the help, life, family, education, and friends. I would like to express my appreciation to my parents, family, wife, and daughters for their unlimited support throughout my life and for offering an appropriate environment allowing for optimal productivity and quality of this research.

I would like to especially thank my supervisor Prof. Abdulmotaleb El Saddik for all his support, enthusiasm, and guidance. Especial thanks to Dr. Julio Valdes, Dr. Jungun Cha, and Dr.Nizar Sakr, and late Dr. Nicolas D. Georganas for their fruitful inputs to this research. Thanks to all staff members of the computer science department at Ottawa university and to my PhD committee members.

Moreover, I am thankful to King Saud University, Department of Computer Science for the scholarship that allowed me to pursue graduate studies in computer science at the University of Ottawa.

In addition, I would like to thank all my friends and family who waited patiently for me to complete my research.

Especially thanks to my colleagues and to those who participated and volunteered in several experiments. Their comments, suggestions, and feedback contributed much to the experiments and to my overall research.

Glossary of Terms

GP Genetic Programming.

Identification The process of distinguishing an unknown subject to reveal hisher identity. A One-to-Many comparison.

KNN K-Nearest Neighbor algorithm

MLP NN Multi-Layer Perceptron Neural Network.

Pre-defined Task A task that has a static target. The order and the steps of the task remain consistant.

Random Movement Task A task that consist of a totally random movement with no specific target. Neither the target nor the path to the target is set.

RUS Random Under-Sampling Technique.

ROS Random Over-Sampling Technique.

Semi-Random Task A task that has a static target. However, the sequence to reach the target might vary.

SVM Support Vector Machines.

Verification The process of asserting the identity towards the claimed one. A one-to-one comparison.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Objective and Motivation

Humans are social beings by nature. People strive to communicate with each other. Families, friendships, education, and businesses are formed based on communication; prompting researchers and inventors to try and answer the human need for socialization and communication. The development of current communication systems started with the invention of radio, text and fax, as such inventions transmit audio. As the development evolved, television was invented to deliver video in addition to audio. However, television, radio and fax were still uni-directional systems rather than interactive systems, until the advent of bi-directional communication means such as the telephone, which carries audio, and teleconference systems that carry audio and video. People around the world adopted the use of teleconference systems that allowed them to see, show, hear, and speak to each other. Research advances in multimedia systems go beyond audio visual communication, to satisfy the remaining human senses by incorporating touch, gustation, and olfaction [26], [25] into multimedia systems. Haptics, derived from the greek word "haptesthai" which means to touch or contact, is a multimedia field dealing with incorporating the sense of touch into multimedia systems. Recent advances in haptics targeted the human nature of communication by incorporating haptics into interpersonal communication systems. Two users connected through a network or through the Internet and who are remotely located can share movements and tactile feelings in order to convey their feelings, emotions, and affections, by adding the sense of touch to audio and video communication [37]. Meeting a person for the first time usually starts with a handshake, and closing a deal starts and ends with a handshake as well (depending

on the culture). Therefore, shaking hands was one of the advancements and targets in haptic interpersonal communications [6]. The ability to hug someone remotely, through the Internet or through the mobile device, is developed [58]. Recently, our lab members [18] developed a system that allows a person to hug someone else remotely, in real time. The other party's video is captured in 3D, and a hug can be sensed by the remote party through a specially designed tactile jacket. As the progress in multimedia systems continue, using such advancements to peoples advantage is the main goal. In contrast, there are people who abuse such technologies, thus harming others by phishing attacks, identity theft, and the use of stolen credit cards. Similarly, in a haptics teleconferencing system, an attacker might impersonate the legitimate user. The attacker might touch others for fun or to intentionally deteriorate the relationship between two legitimate parties. Therefore, advances in security measures that need to be deployed must be made at the same pace as the advances in the correlated multimedia technologies. [8]

There are many security measures related to multimedia systems, starting from encryption algorithms to physical security. However, our concern in this research is the user authentication process which is the process of validating who you are with who you claim to be, as the two parties need to communicate with each other. One common practice is the use of a user authentication to sign-in, by entering a "User Name" and the associated password. This technique is known as Textual Password. For the last few decades, textual password was and continues to be the most common user authentication system deployed. However, users tend to forget, reuse, or share their passwords with friends and family. Consequently, many alternative authentication schemes have been introduced and applied. Token based authentication schemes are commonly used in banking systems as debit or credit cards. This is considered an authentication scheme that is based on what you have. The problem is that tokens might be stolen, forged, or forgotten. Therefore, to solve such problems many token-based systems require an additional measure by combining another authentication scheme such as a textual password or a handwritten signature. Others observe the human uniqueness in genuine physical and behavioural traits, forming the field of biometrics. Biometrics recognition systems vary in the physical or behaviour traits examined, for example fingerprints recognition, hand geometry recognition, palm recognition, face recognition, voice recognition, iris and retina recognition, gait recognition .etc. Biometrics have been utilized for both user verification processes and for identification purposes in many applications, including retrieving health records as well as various forensic purposes. [8]

Most of the traditional authentication schemes such as textual passwords and token-based systems, as well as many biometrics systems such as fingerprints, iris and retina recognition, and palm and hand geometry recognition work by examining the users identity upfront and assuming of the trust of the users identity is established from the beginning to the end of the session. However, such assumptions of trust can no longer be held in cases where a user forgot to end the session (sign-out) and left the system vulnerable for a physical attack by a potential attacker. Moreover, interfering and hacking a legitimate user system is another possibility. As a result, many researchers were motivated to solve such problems and introduced "Continuous Authentication" schemes, where the user is authenticated at the beginning and then continuously during the session, as a solution to "single-point" authentication problems. A continuous authentication method that reflects the static textual password is named "Key stroke dynamics" where the user's keyboard inputs form a behavioural pattern that can be continually verified [44], [46], [50], [56], [55]. As the advances of computer user interactions introduced the mouse and keyboard as input devices, a continuous authentication method of observing the user's mouse movements and forming a pattern as a base for the user's continuous authentication was developed [31], [66]. [8]

With the advancement in multimedia systems and the introduction of haptics to interpersonal communication systems, where users can see, show, hear, tell, touch and be touched, mouse and keyboard are no longer dominant input devices. Touch, speech and vision will soon be the main methods of human computer interaction. Therefore, relying on continuous authentication schemes that meet the interpersonal communication demands is a necessity. Voice recognition, face recognition, and haptic authentication are possible authentication methods. On-line face recognition can verify that the shown video belongs solely to the legitimate user, voice recognition can verify that the sound belongs solely to the legitimate user, and haptics authentication can verify that the touch event was performed solely by the legitimate user. However, voice recognition or face recognition would not verify that the touch belongs to the legitimate user, as a legitimate user might be verified by voice recognition and face recognition and an attacker might interfere by performing the touch action without the permission of the legitimate user. The same scenario would happen if we verify only haptics and not voice. Humans possess the ability to identify the voices and faces of many friends and family members with no effort at all, therefore a change in a person's voice or a change in the shape of their face would be easily identified if both parties know each other well. On the other hand, voice

recognition mostly happens with people that the user knows well, and not with every person he/she meets. Moreover, touch is not shared with everybody, and usually only by a handshake and sometimes hug, depending on the culture. Therefore, an automated haptic authentication system would be required to secure such communication and to prevent any misuse or abuse that may cause misunderstanding, confusion or conflicts between the two parties.

Lately, more specifically November 2011, the topic of continuous user authentication became of central interest to the Defense Advanced Research Projects Agency (DARPA) [35] in the United States. A project named "Active Authentication: Beyond Passwords" presents a futuristic vision that spans from November 2011 to 2015. The project vision targets new authentication modalities for continuous user authentication. The project will try to enhance the classification success rate of each authentication modality, then combine them to create an even more secure environment. Similarly to DARPA futuristic vision, in this research we target the haptic authentication modality, which can be seen as an active authentication that will be combined with other authentication modalities in the near future.

In order to reach our target of using haptic information to identify and verify a user's identity based on continuous yet random movements, we divided our research into three stages of experiments. Each stage or level represents a succession from the previous one. Fig. 1.1 illustrate the three main stages and their attributes.

Our starting point is improving the classification and verification accuracy of the handwritten signature with haptic information, though it is not our main goal. A users signature is usually only requested once, for example when someone issues a cheque or signs into a system (e.g. an email system). Any additional requests for a handwritten signature would hinder the user and as a result decrease the usability of the system, for example in an email system based on handwritten signatures where a signature is required every time the user reads an email message. No typical user would use such a system to browse and read hundreds and sometimes thousands of email messages, even if this level of security might be necessary in a system that requires a top level of security.

In order to escalate the complexity of our proposed system, we conducted experiments to observe user's drawings. Sketching a picture can be performed in several different

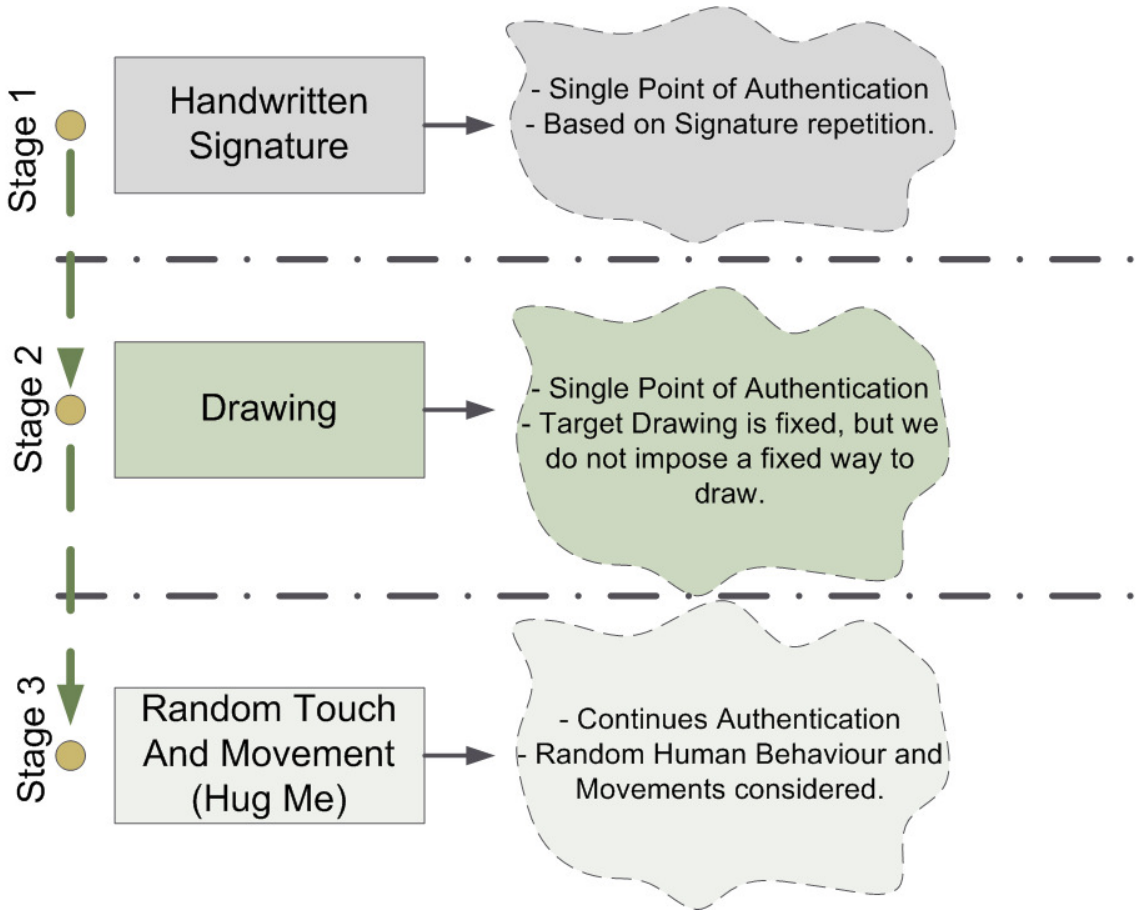


Figure 1.1: Research Targets divided into three main stages

ways, even by the same user. Details might be omitted in one trial, while other details might be emphasized. Although the target is the same, the result varies from one trial to another. We therefore designed and developed experiments that can be characterized with "semi-random" movements, though no such system would be applied in real life due to usability issues. However, such a stage is considered important for our research in order to raise the level of complexity of the proposed system, from a system that captures signatures performed with the intention of repeating (literally) the authorized signature, to a system that observes and authenticates users based on random movements and behaviours. Semi-random movement system that is considered as a single point of authentication system, similar to that of the previous stage.

Our third and last stage is based on fully random movements and behaviours, where there are no restrictions on the user's movements. In this stage, a continuous haptic authentication in an interactive multimedia system is developed. The main concern is to prevent the interactive multimedia system from intrusion during the interaction session or when the user leaves the desk without logging out. Moreover, the proposed continuous authentication haptic information system should be coupled with a single point of authentication system in order to prevent a postponed decision of "reject" signaled by the continuous authentication system.

In our research, haptic information such as position, velocity, and force of a haptic device tool-tip has been utilized to perform the identification and verification of users. Analyzing the contribution of each haptic feature toward user authentication is another objective of this research.

The following are our objectives for this thesis:

- Enhancement of user identification and verification success rates based on handwritten signatures with haptic information.
- Examining the feasibility of user identification and verification based on user drawings with haptic information.
- Analyzing the contribution of each haptic feature toward user authentication.
- Examining the possibility of utilizing haptic information to build a continuous user

authentication system.

1.2 Thesis Contribution

The proposed user authentication technique facilitates users' identifications and verifications based on their behavioral properties. Users movements or haptic information such as position, force, and angular rotation of the haptic device tool-tip are recorded and processed. Consequently, the analysis of the haptic information allows the system to determine whether or not the user is legitimate. Our system does not capture any physical properties of the user. For this reason, user privacy concerns, which are a major drawback of traditional biometric systems that rely on physical traits, is decreased with the utilization of a biometric system that relies on behavioral properties.

The contributions of this thesis are:

1. Design and development of the PROFECE system for user authentication based on haptic information.
2. Design and development of feature reduction algorithms based on genetic programming.
3. Development and utilization of several methods to solve the unbalanced data set problems such as the weighting factor (Section 3.5.2), random minority oversampling (ROS), random majority undersampling (RUS) in combination with several classical classifiers.
4. Design and development of fitness functions that guide the evolutionary computing (namely genetic programming) towards analytical functions that solve the unbalanced data set problem.

The proposed solution in this thesis targets continuous user authentication where the system continuously authenticates users, rather than a single point authentication procedure in which user authentication is only required at the beginning of a session. The need for continuous authentication increases as users and devices are interconnected from all over the world. Thus, hackers attacks can occur not only at the beginning of a session, but during a running session as well.

A series of user experiments is performed, and analyzing users haptic information and the resulting validation and identification success rate is a crucial measurement in our research.

1.3 Thesis Organization

In this chapter, the objective and motivation for the current research is presented and the three stages and milestones of our research are introduced. We follow with the thesis contribution, the thesis organization, and scholarly activities resulting from this thesis are listed.

Chapter 2 presents the existing haptic authentication schemes in depth. It starts with a discussion of the general authentication systems and its taxonomy. This section starts with textual password and then examines biometric recognitions and token-based authentication schemes. Finally, the section explores various haptic authentication schemes in detail and present the related work on the unbalanced dataset problem.

Chapter 3 Showcases the design of PROFECE our Proposed framework for haptic-based user authentication. It discuss the methodologies and techniques applied in our research, and presents haptic feature dimensionality reduction approach and time-varying approach followed in this research which informs what combination of algorithm are used. Moreover, it illustrate a road map on where each methodology can be applied.

Chapter 4 presents the PROFECE Verification with predefined tasks. It presents the first step namely the virtual check application. An experiment where a predefined task is required. It presents several classifiers, feature selection methods, and several solution to enhance identity verification based on haptic information.

Chapter 5 discusses Semi-Random Tasks in PROFECE where users are verified based on their drawings. The level of user movement increased in this step; however, the target landscape picture is still predefined.

Chapter 6 presents PROFECE for random tasks namely in haptic-enabled interpersonal communication system. A task that provide a fully user movement freedom. User identification and verification are discussed and the experimental results are presented.

Chapter 7 Concludes and discuss our future work.

1.4 Publications Resulting from this Research

- A Synchronous Interpersonal Haptic Communication System. United States Patent US8294557 B1, 12/481,274, October 2012.
- F. A. Alsulaiman, J. Cha, and A. El Saddik. User Authentication Using Haptic Information in Drawing, In progress.
- F. A. Alsulaiman, N. Sakr, J. J. Valdes, A. El Saddik. Identity Verification based on Handwritten Signature with Haptic Information using Genetic Programming. ACM Transactions on Multimedia Computing, Communications and Applications (ACM TOMCCAP), 9(2), May 2013.
- F. A. Alsulaiman, J. J. Valdes, A. El Saddik. Identity Verification based on Haptic Handwritten Signature: Genetic Programming with Unbalanced Data. IEEE Symposium on Computational Intelligence for Security and Defence Applications (CISDA), Ottawa, ON, Canada, July 2012.
- N. Sakr, F. A. Alsulaiman, J. Valdes, A. El-Saddik, N. D. Georganas. Feature selection in haptic-based handwritten signatures using rough sets. In Proceedings of the 2010 IEEE World Congress on Computational Intelligence (WCCI 2010), Barcelona, Spain, July 2010.
- N. Sakr, F. A. Alsulaiman, J.J. Valdes, A. El Saddik, N.D. Georganas. Relevant Feature Selection and Generation in High Dimensional Haptic-based Biometric Data. Proc. International Conference on Data Mining (DMIN'09), CSREA Press, Las Vegas, NV, USA, July 2009.
- F. A. Alsulaiman, N. Sakr, J. J. Valdes, A. El Saddik, N.D. Georganas. Feature Selection and Classification in Genetic Programming: Application to Haptic-based

Biometric data. IEEE Symposium on Computational Intelligence for Security and Defence Applications, Ottawa, ON, Canada, July 2009.

- R. Iglesias , F. A. Alsulaiman, M. Orozco, J.J. Valdes, and A. El Saddik. Characterizing Biometric Behavior through Haptics and Virtual Reality. In 42nd IEEE International Carnahan Conference on Security Technology, Prague, Czech Republic, 13-16 October 2008.

- F. A. Alsulaiman, J. Cha, and A. El Saddik. Preliminary Results of a User Identification Using Haptic Information. IEEE International Conference on Virtual Environments, Human Computer Interfaces, and Measurement Systems, (VECIMS), Istanbul , Turkey, 14-16 July 2008.

- F. A. Alsulaiman, J. Cha, and A. El Saddik. User Identification based on Handwritten Signatures with Haptic information. In Proc. of the EuroHaptics 2008 conference, Madrid, Spain, 12-14 June 2008.

Chapter 2

Related Work

2.1 Haptic Interpersonal Communication

Research advances in multimedia systems go beyond audio visual communications, to satisfy the remaining human senses by incorporating touch, gustation, and olfaction [26], [25] into multimedia systems. Haptics is a field that incorporates the human sense of touch into multimedia systems. Recent advances in haptics targeted the human desire for socialization with the incorporation of haptics into interpersonal communication systems. Two users connected through a network or through the Internet and who are remotely located can share movements and tactile feelings in order to convey their feelings, emotions, and affections, by adding the sense of touch to audio and video communication [37]. Haans and Ijsselsteijn [37] defined mediated social touch as "the ability of one actor to touch another actor over a distance by means of tactile or kinesthetic feedback technology". Currently, many researchers are working on finding a proper design for haptic devices that facilitate a proper haptic interaction among remote parties. For example, when people meet for the first time, a handshake is usually the first action taken while introducing oneself verbally. Thus, shaking hands was one of the targets in haptic interpersonal communications [6]. More specifically, alhalabi and Horiguchi[6] introduced tele-shake where two people can shake hands virtually through a virtual environment by using SensAble Technologies PHANTOM device[68] as in figure 2.1. DiSalvo et al.[22] observed several images of interpersonal communication through touch and designed "The Hug", a robotic toy that can be hugged. Moreover, a voice tag is stored to indicate for whom the hug is intended. By providing the name while hugging "The Hug", the other party will receive the hug in the context of a "The Hug" vibration. Moreover, the

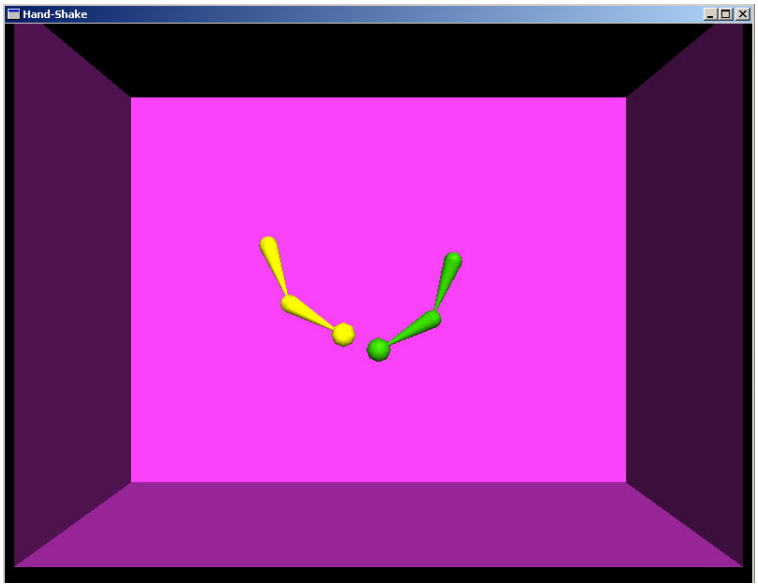


Figure 2.1: A snapshot of a tele-handshake system where it uses customized virtual environment for handshake action[6].

feeling of a hug that can be sensed from a remote partner through a vest was introduced by Mueller et al.[58]. Mueller et al.[58] designed a koala toy that acts as an input device to send hugs to the partner, who is wearing an inflatable vest that inflates to imitate the feeling of hug[58]. However, the hug is only half sensed, since the receiving party can feel the hug sensation, yet not the sender. Recently, our lab members [18] developed a HugMe system that allows the user to hug someone, remotely and in real time. It uses a haptic device as an input and the other party's video is captured in 3D. The hug can be sensed through a specially designed tactile jacket as in figure 2.2.

2.2 Haptic Authentication

Orozco and El Saddik [61] studied the quantification and recognition of human movement patterns through haptic applications. They [61] conducted their study using the following three haptics applications:

- solving a maze application (figure 2.4).
- signing a virtual cheque (figure 2.5).

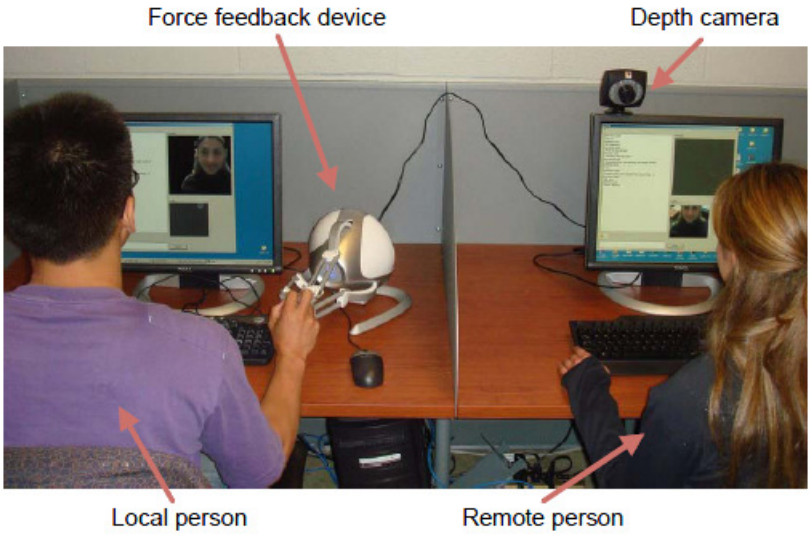


Figure 2.2: A HugMe system where one user is sending hugs through the haptic device and the other party wearing a haptic jacket allowing her to receive hugs remotely [18].

- dialing a virtual phone (figure 2.3).

Moreover, Orozco and El Saddik [61] generated a set of 35 features per trial, and applied the k-means algorithm for feature selection. They used the proximity measure based on euclidean distance and cluster criterion in order to quantify the similarities of any two feature vectors processed. In addition, they applied the majority classifier [51] and considered a test data belonging to the legitimate user only if more than half of the tested features were considered genuine. Orozco and El Saddik [61] reported an equal error rate (EER) between 6% and 9% for the virtual check application, and between 15% to 22% for the maze and virtual phone applications respectively.

Orozco et al. [60] conducted an experiment on the virtual maze (figure 2.4) on 22 subjects. Every user was asked to solve the maze ten times. They [60] asked the user to navigate and solve the maze which contains sticky walls. Before starting to record the user's data, they briefed every user and requested that the user solve the maze a couple of times, to ensure that they knew how to solve the maze and that they were familiar with the haptic device. First order analysis, dynamic time warping, and a single parameter hidden markov models (HMM) were used for user authentication. In the case of HMM, they divided the users navigation into multiple stages in which strokes are divided into

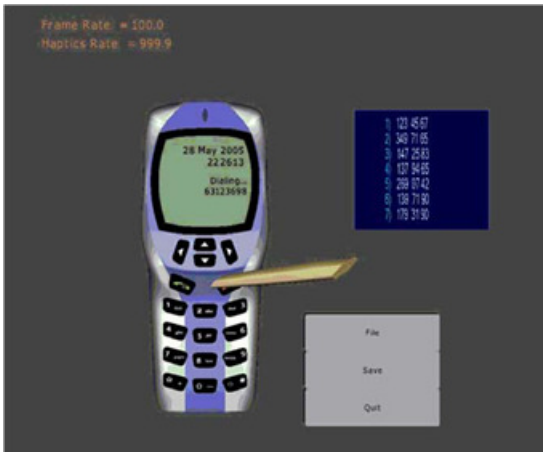


Figure 2.3: Screen shot of phone application where user is required to dial the numbers written on the right side[59].

segments. They reported [60] ,after discarding the first 5 trials from the training set, a probability of verification (PV) of 78% at 25%FAR using the spectral analysis technique and a PV of 60.1% using time warping technique. However, when the whole training set is considered, a probability of verification of 67% and 49% using spectral analysis and time warping respectively, were achieved. Moreover, in the case of HMM, they achieved an average PV of 66% considering the top 2 log likelihood for four users, and a PV of 41% when the top 1 is considered [60]. Top 1 considers a correct classification when the top predicted user matches the legitimate user, while top 2 is when one of the two predicted users match the legitimate user.

El Saddik et al. [27] further investigated the user identification and verification based on the virtual check application and the maze application. Twenty-two subjects conducted the experiment with the virtual maze, while 16 subjects performed the virtual check experiment. They [27] studied the effects of the different features, through the use of relative entropy, by comparing the interpersonal distribution to the intrapersonal distribution. As a result, an analysis of haptic features such as displacement in x, y, and z direction, weighted velocity, styles velocity, torque in x, y, and z direction, force and angular orientation were presented in terms of information content in bits. They observed that force followed by torque revealed most of the information about users [27]. Therefore, their feature extraction process considers the force and torque features

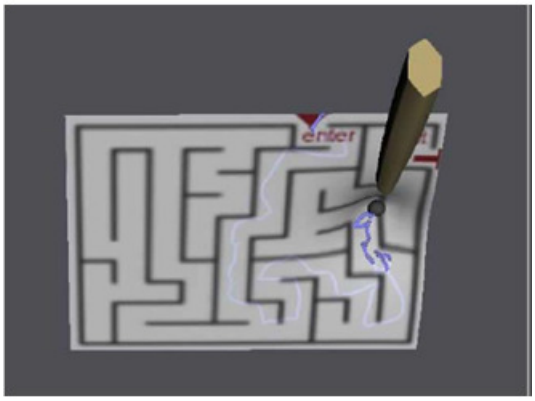


Figure 2.4: a screenshot of the maze application. A user is required to solve the maze starting from enter point and follows a path to the exit point[59].

to build the user's profile. Thus, six features are extracted, and for each each (such as force in x direction), a hamming window with a length of 256 is applied. Consequently, a biometric profile matrix of 6×256 is formed. To compare a template biometric profile matrix to the one presented as a test, a quantitative match score is used for classification, as follows:

$$MS = \ln \left\{ \sum_{i=1}^{6} \sum_{j=1}^{256} (\|d_{1,i,j}\| - \|d_{2,i,j}\|)^2 \right\} \tag{2.1}$$

Where the summation covers the six features $i = 1..6$ and the profile windows of 256 ($j = 1..256$). $\|d_{1,i,j}\|$ is the complex norm of the i, j^{th} element[27]. A low match score (MS) value means a high match between the users biometric profile template and the tested one. A threshold τ on the resulted MS values was defined as a fine line to determine whether the decision is to accept or reject[27]. They [27] reported a user identification success rate as PV of 71.2% at FAR of 25.5% and when the window size was increased to 512, PV improved to 75.5% at FAR of 24.7%. Moreover, based on the virtual check application and using the first eight trails, a PV of 50% at FAR of 25% was reported. As for user verification, they included additional haptic features such as average angular orientation and total time taken to complete the maze challenge. They added the following three constraints in the verification process:

- a predetermined user-dependent threshold rather than a universal threshold.

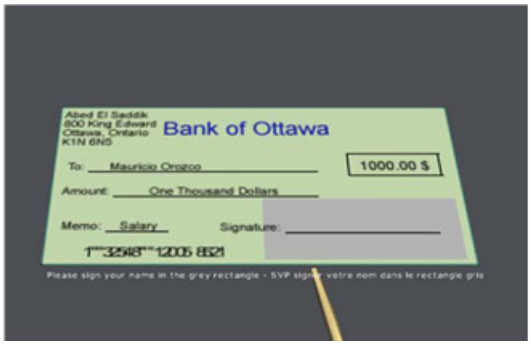


Figure 2.5: A screen shot of the virtual cheque application where a user is required to sign a cheque [59].

- a constraint on time completion, by enforcing a total time of completion of no more than 26.4% of the profiles average time. Therefore, a user who takes a longer period of time (more than 26.4% of the average completion time when creating the user's template) will be rejected.
- a constraint on average angular orientation, by enforcing a less than 20% difference to the original template.

Based on the above constraints, a PV rate of 95.4% at FAR of 4.5% is reported.

Malek et al. [54] proposed an approach to utilize haptics for shoulder surfing attack prevention. Their work [54] is based on the graphical password scheme presented by Tao [71] where users draw a secret and the grid points that intersect with the user's strokes form the user's secret or passgraph. Malek et al. [54] augment haptics by measuring the user's force exerted while connecting the grid points as either high pressure or low pressure, as shown in figure 2.6. The objective of the addition of haptics is to prevent shoulder surfing attacks, where an attacker observes, or records by camera, the legitimate users graphical password, or any other type of authentication presented. The presented scheme partially prevents shoulder surfing attacks since the attacker can still watch the legitimate graphical password presented, yet the force applied is hidden from him[54]. In addition, Orozco et al.[63] utilized Artificial Neural Network (ANN) and Nearest Neighbor (NN) for passgraph recognition. They reported a PV of 92% and of 90% for NN and ANN, respectively.

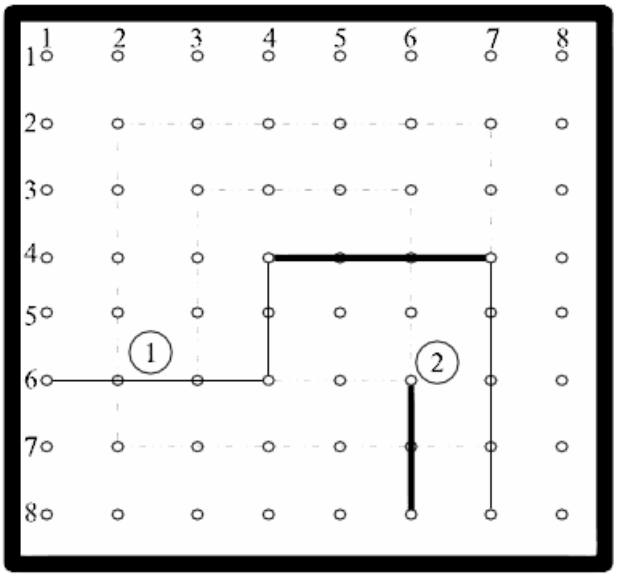


Figure 2.6: An illustration of the passgraph scheme with haptic support where a user is required to draw a passgraph. The bold lines represent a high pressure line drawing by a user while remaining lines represent Low pressure lines. [54]

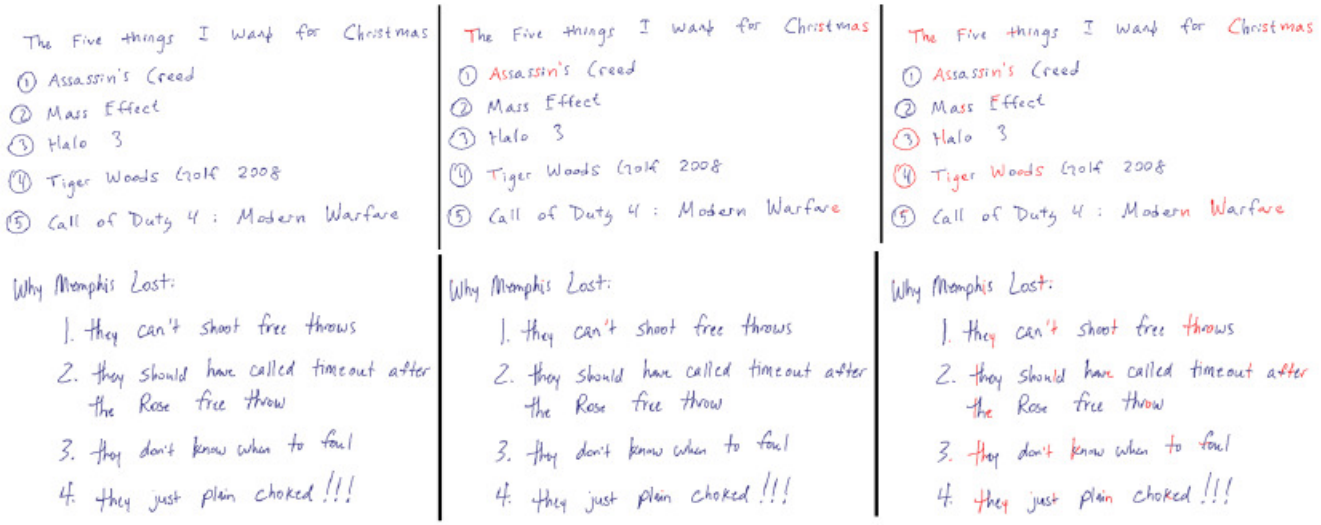


Figure 2.7: A snapshot of the context free differentiation system through sketch. Blue color strokes indicates correct discrimination while red color strokes indicates incorrect classification.[28]

Eoff and Hammond [28] presented a sketch identification system for a multi-user collaborative system where a user can write down a sentence and another user can sketch other strokes in the same environment. The identification purpose is motivated by recognizing the contribution of every user that participated in the drawing. The system uses the pen's tilt as well as the applied pressure and the velocity as features to discriminate users. The introduced system is validated via an experiment where users are asked to write down a list of what to do during the experiment day. Figure 2.7 illustrates an example of the proposed experiment for this system. Based on ten users, Eoff and Hammond [28] achieved an accuracy of 83.5%. The proposed system experiment limited the stroke movement randomness to the 52 letters and ten numbers (total of 62), Since every sentence consist of words, and every word consists of group of letters and/or numbers.

2.3 Continuous Authentication

Nowadays, in almost every system, the first action to be taken to gain control of a system is user authentication. It starts mostly by requesting a username, a token (like a key or a card), or sometimes a fingerprint or any other authentication method. The main characteristics of such authentication processes are:

- Authentication process occurs only at the beginning and prior to gaining control of the system.
- A short time frame is required for the authentication process.
- No further re-authentication is required.
- User is considered legitimate as long as the initial authentication process succeeded.

Those characteristics form what we recognize as "one-time-frame authentication systems". Most existing authenticating systems fall under the umbrella of one-time-frame authentication systems. The trust fallacy in this situation occurs for the following reasons:

- A legitimate user might deliberately allow others to use the protected system on his behalf.
- A legitimate user might share his key intentionally, or mistakenly with others.

- A legitimate user might forget to shut down or end the authenticated process.
- An attacker might impersonate the legitimate user to access the protected resources.

As a solution, and in order to mitigate the risk exposure resulting from the use of one-time-frame authentication systems, continuous authentication systems were proposed by many researchers [50, 56, 55, 31, 66]. Continuous authentication systems observe the users behaviour at run time and continuously authenticate the user who is currently operating the system, rather than just authenticating the user who was operating the system at start time. The following characteristics describe continuous authentication systems:

- Observe the users behaviour and patterns, in terms of usage or in terms of physical movements.
- The authentication process operates as long as the user is using the system.
- An alert or the rejection of a user can occur in the middle of an active session.

Moreover, the following characteristics represent an ideal continuous authentication system:

- An alert or a reject should not hinder the usability of the protected system.
- The continuous authentication system should have a minimal overhead.
- User is aware of the benefits and of the use of the authentication system.

Continuous authentication systems based on keystroke dynamics utilize the user's keystroke rhythms and typing patterns to differentiate between users. Keystroke dynamics is transparent and does not impede the usage of the protected system. Moreover, it does not require any additional device other than a keyboard which is already used in many systems. Lee et al. [50] proposed a system that considers keystroke dynamics by constructing a timing vector, where a timing vector considers the time needed to type a letter and the latencies between typing two letters, as in figure 2.8. The system proposed [50] is based on ellipsoidal hypothesis spaces and utilizes genetic algorithms to optimize the process of finding the proper hypothesis space [50]. Jiang et al. [44] proposed a

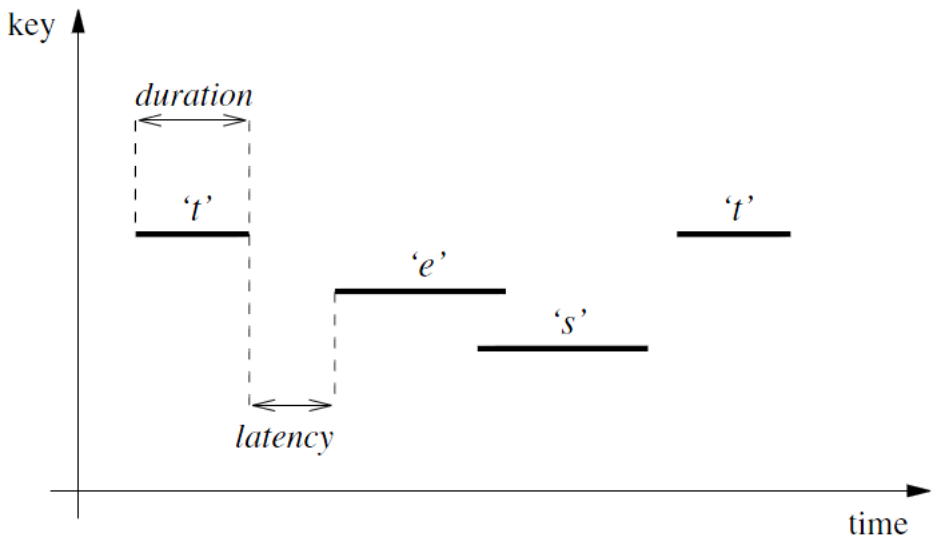


Figure 2.8: Timing Vector correspond to typing "test" where duration represents the pressed letter time, and latency represent the time required to press the next letter [50].

keystroke statistical learning model for user authentication. Their proposed statistical model is based on Gaussian modeling and the Hidden Markov Model (HMM). They [44] achieved an error rate of 2.54% at best. Monrose and Rubin [56] discussed the keystroke dynamic and made a fine, precise, and short comparison between textual passwords and keystroke dynamics as "Not What you Type But How you Type"[56]. They developed a toolkit for the authentication of keystroke dynamics using euclidean distance measures, non-weighted probability, and weighted probability as classification methods [56]. Monrose et al. [55] proposed a keystroke dynamics scheme that is combined with the textual password for password hardening.

As the development of computer interaction continues, the use of the mouse was introduced as an input device. Gamboa and Fred [31] proposed an identity Authentication system based on human computer interaction behaviour through mouse movements (see Fig. 2.9). Similarly, Pusara and Brodley [66] proposed a system for user re-authentication via mouse movements that considers mouse movements and clicks in user authentication.

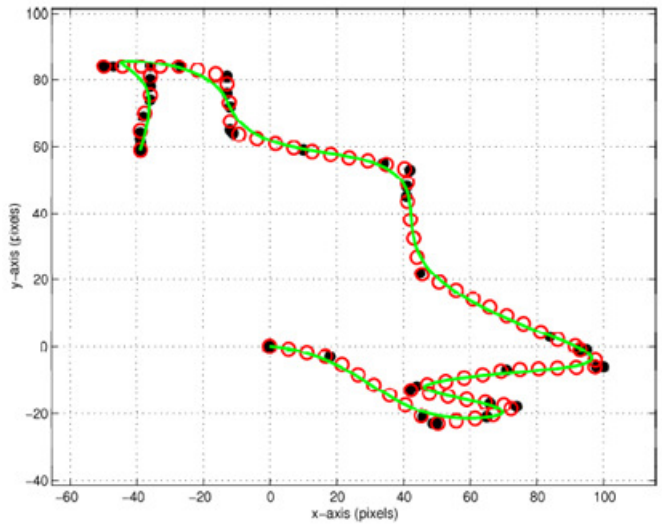


Figure 2.9: Mouse movement represented in x, y signal. Black dots represent the input sampled point while the white dots represent linearly interpolated points. [31]

2.4 Unbalanced Data Set Problem

The study and analysis of user behaviour for the purpose of identity verification requires the collection of user data beforehand. Moreover, as the information collected for each user might vary, the identity verification process might encounter the unbalanced data set problem. In this section, we will discuss the solutions to this problem from a general perspective, where the solution can be applied to various classical classifiers. Then we will discuss solutions that are specific to evolutionary computing.

2.4.1 Unbalanced Data Set Problem from a General Perspective

As in real life, we often encounter problems that require us to find and observe most of the present data to form a decision. Sometimes there are cases where we cannot obtain enough information, but a decision must still be made based on the available information. The uneven distribution of instances presented might affect the decision made by the classifier. A classifier might estimate an accurate prediction for a class that is under-represented. Therefore, many solutions have been proposed to solve the unbalanced data set problem. Kubat and Matwin [49] addressed what they consider "the

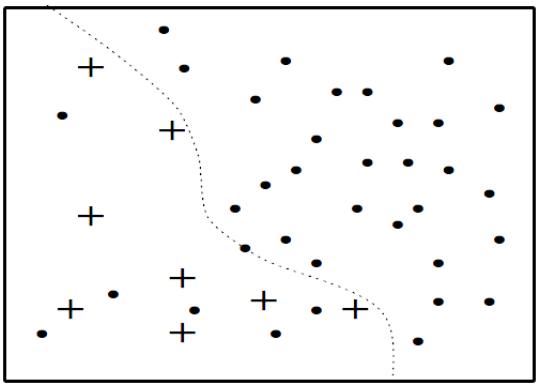


Figure 2.10: Instances in training set where + represent the minority class examples, the dotted line represent the border line. [49]

curse of imbalanced training sets" and proposed a "One-Sided Sampling (OSS)" where sampling occurs literally on one side only. OSS [49] removes redundant and border line instances. The detection of noisy examples and border line examples is based on Tomek links [72]. Tomek links works as the following: assume there are two examples x and y that belong to two different classes. Assume $\delta(x, y)$ represents the distance between x and y. (x, y) is considered a Tomek Link only if z does not exist where $\delta(x, z) < \delta(x, y)$ or $\delta(y, z) < \delta(y, x)$ [72] [49].

Another solution named "SMOTE" is proposed by Chawla *et al.* [20]. SMOTE stands for "Synthetic Minority Over-sampling TEchnique" [20]. SMOTE introduces a synthetic instance to over-sample the minority class. The algorithm works by picking a minority class instance and considering its neighbours. It then calculates the difference between the two instances by considering the feature vector. We can see the difference as a distance between the two points. SMOTE randomly selects a point along the line between the two points [20]. Moreover, SMOTE combines their synthetic over-sampling of the minority class with a random under-sampling of the majority class.

Another approach used to solve the problem of an unbalanced data set is by assigning costs to examples, with a higher error cost for minority class and a lower cost for majority class, as in Pazzani *et al.* [65].

Japkowics and Stephen [43] studied the unbalanced data set problem by raising three main questions [43]

- What is the nature of the class unbalance problem?
- How do unbalanced data set solutions compare?
- Is the classical classifier accuracy affected by the unbalanced data set problem?

They outlined the following four main factors that effect the problem of unbalanced data sets [43]

- The degree of class unbalance.
- The size of the training sets provided.
- The complexity of the resulting concept.
- The type of classifier used.

A comprehensive experimental study that covers 35 datasets, a set of seven techniques to solve the unbalanced data set problem, and eleven classifiers, was conduced by Hulse *et al.* [41]. They noted that some solutions of unbalanced data set problems work well with specific learners while some learners might not be improved by the usage of a specific solution to the unbalanced data set problem. Hulse *et al.* [41] concluded that random sampling techniques such as Random majority UnderSampling (RUS) or Random minority OverSampling (ROS) outperformed the intelligent sampling techniques such as SMOTE[20] and OSS [49].

2.4.2 Unbalanced Data Set Problem from a Genetic Programming Perspective

Patterson and Zhang [64] in addition to Bhowan *et al.*[11] introduced an unbalanced data set solution from a genetic programming perspective; they showed that utilizing overall accuracy as a fitness function evolves classifiers with discrimination bias towards the majority class. As a solution they [11] introduced four GP fitness functions that consider the performance of the majority class and the minority class separately. The four fitness functions introduced are:

$$\text{Fit1} = \sqrt{\frac{hits_{min}}{N_{min}} \times \frac{hits_{maj}}{N_{maj}}} \tag{2.2}$$

where $hits_{min}$ are the number of correct classifications of the minority class, $hits_{maj}$ represents the number of correct classifications of the majority class, N_{min} is the total number of minority class instances and N_{maj} is the total number of majority class. Equation 2.2 utilizes the geometric means of the two objectives, namely maintaining the classifier performance for both the majority class and the minority class.

The second fitness function introduced by [11] is conceptually based similarly to Equation 2.2 with the addition of a third objective which is maintaining an overall classification accuracy rather than just focusing on the classification accuracy of the majority and minority classes. Thus, [11]

$$\text{Fit2} = \frac{hits_{min}}{N_{min}} + \frac{hits_{maj}}{N_{maj}} + \frac{hits}{N} \tag{2.3}$$

where $hits$ represents the overall number of correct classifications and N represents the total number of instances. The third fitness function introduced by Bhowan *et al.*[11] dives in-depth into the classifier results to differentiate two classifiers with the same classification accuracy for minority and majority classes, but with different classification models. [11]

$$\text{Fit3} = \frac{hits_{min}}{N_{min}} + \frac{hits_{maj}}{N_{maj}} + (1 - Rng_{min}) + (1 - Rng_{min}) \tag{2.4}$$

where

$$Rng_c = \frac{Pout_c^{max} - Pout_c^{min}}{2} \tag{2.5}$$

where the Rng_{min} and Rng_{maj} represents the ranges of incorrectly classified instances evaluated in equation 2.5. $Pout_c^{maj}$ and $Pout_c^{min}$ represent the smallest and the largest incorrectly classified instances from class c.

The forth fitness function developed by Bhowan *et al.*[11] is formed over the Wilcoxon-Mann-Whitney (WMW) statistic. WMW is a shortcut to overpass the calculation of the area under the curve (ROC) by computing an equivalent estimator (WMW) as the following [11]

$$\text{Fit4} \ = \frac{\sum_{i=0}^{N_{min}} \sum_{j=0}^{N_{maj}} I(x_i, y_i)}{N_{min} \times N_{maj}} \tag{2.6}$$

$$\text{where } I(x,y) = \begin{cases} 1 \ x > 0 \text{ and } x > y \\ 0 \text{ otherwise} \end{cases}$$

Similarly, Patterson and Zhang [64] introduced the following two fitness functions

$$\text{FitPat1} \ = \frac{\frac{hits_{maj}}{N_{maj}} + \frac{hits_{min}}{N_{min}}}{2} \times 100\% \tag{2.7}$$

The second fitness function proposed by Patterson and Zhang [64] is

$$\text{FitPat2} \ = \frac{\left(\frac{hits_{maj}}{N_{maj}}\right)^2 + \left(\frac{hits_{min}}{N_{min}}\right)^2}{2} \times 100\% \tag{2.8}$$

The two proposed fitness functions [64] resulted in better minority class classification in comparison with the use of accuracy measures for classification based on SPECT heart data [12],pima Indian diabetes data(PIMA)[12], and synthetic data sets.

Bhowan *et al.*[11] applied the developed four fitness functions equ.(2.2)(2.3)(2.4)(2.6) on three uneven distributions of instances; specifically, SPECT and YEAST from the UCI repository [12] and face image data set from the Center for Biological and Computational Learning at MIT [70]. As a result, fitness function 2.4 achieved the highest accuracy from the minority class perspective, and the fitness function 2.3 achieved the highest overall accuracy.

Gathercole and Ross[32] described three subset selection methods namely Dynamic Subset Selection (DSS), Historical Subset Selection (HSS) and Random Subset Selection

(RSS) to reduce the number of GP's tree evaluations. DSS is based on two factors: the selection of difficult cases and the disused cases (age factor). HSS relies on previous GP runs to choose a static subset through all GP generations. RSS is based on the selection of a random subset in every generation.

Based on the concept of the increased probability of producing degenerate classifier models as long as it is based on an unbalanced data set [69][24], Doucette and Heywood[23] designed a robust classifier that is based on two concepts. First, building an appropriate exemplar selection technique. Second, designing an efficient fitness function. They proposed a subset exemplars selection technique "Simple Active Learning Heuristic (SALH)" where the fitness function is evaluated with uniform distribution of the exemplar from the majority class and the minority class [23]. The fitness function is based on the Wilcoxon-Mann-Whitney(WMW) statistics and used as an efficient estimator, instead of calculating the AUC through the calculation of ROC [75].

2.5 Conclusion

The process of user authentication based on haptic information is still an open problem that needs to be addressed. Most of the previous work targets a pre-defined task for identity authentication. Haptic applications such as military simulation training, medical surgery operations, rehabilitation, etc. are mostly real time systems that are dynamic and continuous. Pre-defined tasks for user authentication is no longer sufficient to ensure user identity. Therefore, continuous authentication is an additional measure to ensure user identity during an active session in addition to ensuring the identity at the start of the session.

The process of user authentication based on haptic information consists of pre-processing, feature selection and generation, classification and evaluation. Each step can be addressed based on various techniques and algorithms.

Chapter 3

PROFECE: Proposed framework for haptic-based user authentication

In this chapter, we present the design of the proposed framework as well as the methodologies we are going to apply in order to achieve a higher accuracy in user authentication. Moreover, the significance of original haptic features on the classification process will be investigated. At the beginning of this chapter, an introduction that describes the overall view of the proposed system and the layout of the methodologies' utilization is formed. We follow with a description of the two main authentication targets. The description follows the generic process of user authentication. Then, the problem of unbalanced data is described along with the methodologies applied for their utilization in the generic classification process.

3.1 Overview of the Proposed System

In this section, we explore and present the methodologies that we are applying to solve the problem of user identification and authentication, using haptics data. Moreover, in this section, we also form an overview of the system to illustrate what methodology is applied and when. Fig. 3.1 illustrates a general overview diagram of the proposed system to solve the problem of user identity verification and identification. We call the system PROFECE as it consists of four steps: pre-PROcessing, FEature extraction, Classification, and Evaluation. A pre-processing step is utilized to re-evaluate instance distribution by analyzing the occurrences of instances belonging to one class over another (unbalanced data set problem). A nearest neighbour up-sampling or down-sampling might be required

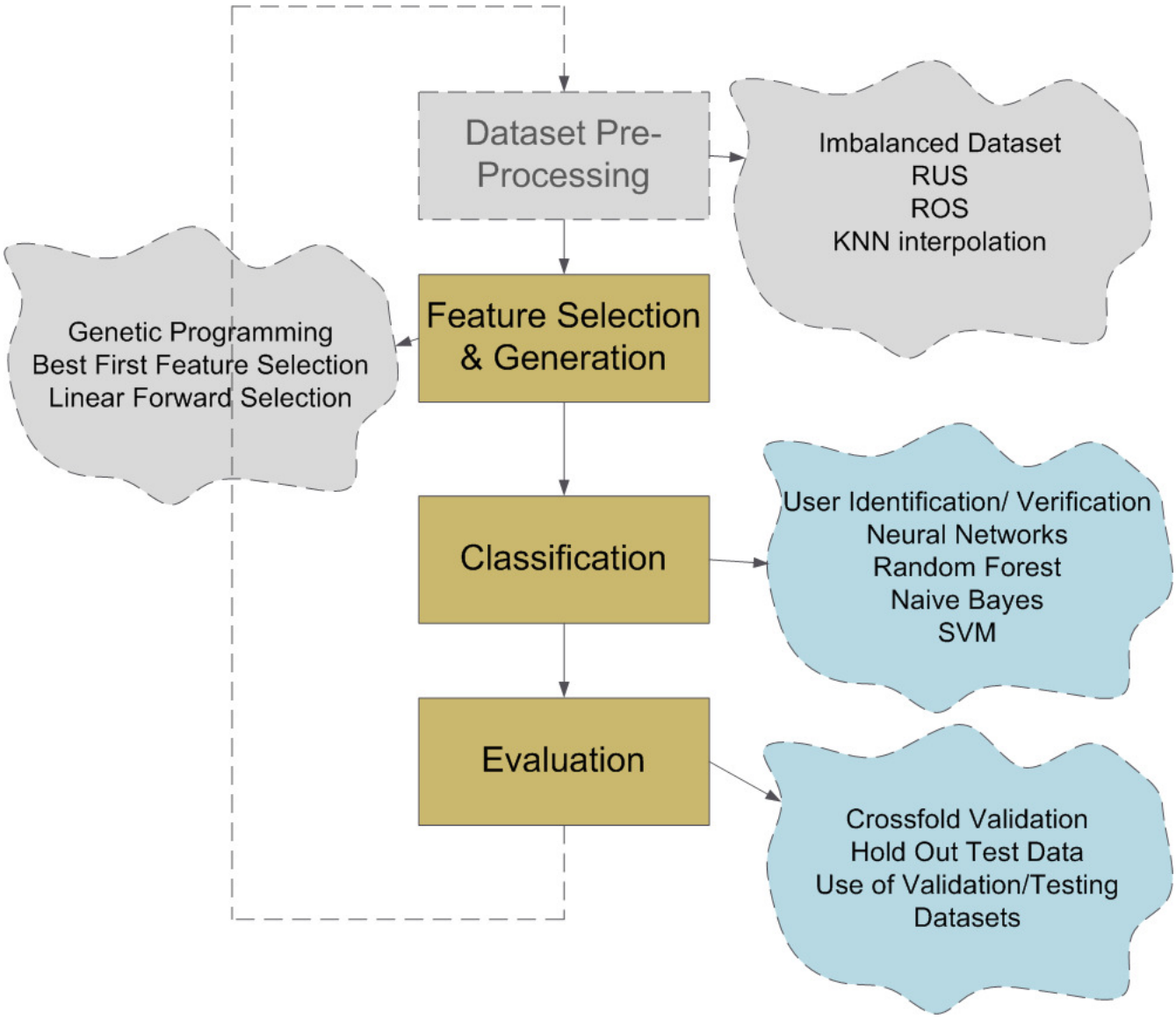


Figure 3.1: A flowchart diagram for combining several methodologies for the purpose of user identification and authentication based on haptic information in PROFECE

based on the data set presented. Finding and quantifying the haptic features contribution toward user authentication is a result of the feature selection and generation process, as several algorithms are used to search the feature space for irrelevant features. In this process, genetic programming is exploited and compared to other classical methods of feature selection such as best-first feature selection and linear forward selection. Many classification algorithms have been utilized in addition to Genetic Programming (GP) such as Neural Networks (NN), Random forest, Support Vector Machines (SVM), and K-Nearest Neighbor (KNN). GP as a classifier produces analytical functions that explicitly derive generated features and can therefore be viewed as a white box classifier. Thus, empowering us with the capability of measuring the contribution of every haptic feature, while the usage of the remaining classifiers result in models that are not explicitly derived from the feature space.

As for the proposed system evaluation and testing, a hold-out data set has been utilized in addition to the use of cross-fold validation. In order to make a clear distinction between the existing work utilized by PROFECE and the parts that we have proposed, we will first go through the methodologies utilized by PROFECE. Then, we will present the methodologies that contain our proposed contributions. At the end we will show how we utilize several methodologies to form PROFECE.

3.2 User Verification Vs User Identification

There are two main targets of user authentication. The first is user verification, where it verifies a user versus what the user claims to be. Thus, it is a one-to-one comparison. User verification is mostly used to sign-on a system. A decision of "YES" means that this is the right user and a "NO" decision informs the system of an illegitimate user trial. However, user identification answers the question who is this or what is this, thus it is a one-to-many problem. An answer would be, for example, this is James Bond or this is an unidentified person or object. User identification is usually used on video surveillance, and forensics is usually a main application of identification. The differences between these two concepts is crucial in selecting the methodologies for our proposed system and in identifying its requirements.

3.3 PROFECE Background

In this section, we show the methodologies utilized by PROFECE without any contributions.

3.3.1 Data Pre-Processing

The data pre-processing step starts by examining our target application to determine whether it is a user verification or a user identification one. For user identification, instances are assigned to the corresponding subject class; in user verification, we present the data as a binary problem. In addition, Nearest Neighbor Interpolation and solving the unbalanced data set problem are part of the data pre-processing step.

Nearest Neighbor Interpolation

Nearest Neighbor Interpolation is a well known method of utilizing known samples to interpolate synthetic samples. The use of the Nearest Neighbor Interpolation in our work is to either up-sample or down-sample the feature space, if needed. The Nearest Neighbor Interpolation is as follows:

$$\hat{y} = f(knn) \tag{3.1}$$

thus,

$$\hat{y} = \frac{\sum_{i=1}^{knn} \frac{y_i}{(d_i)^p}}{\sum_{i=1}^{knn} \frac{1}{(d_i)^p}} = \frac{\sum_{i=1}^{knn} w_i \times y_i}{\sum_{i=1}^{knn} w_i} \tag{3.2}$$

where $w_i = \frac{1}{(d_i)^p}$,$p = \{1, 2\}$, and knn is the number of neighbours, w represents the weight that reflects the effect of the neighbour y_i on $\hat{y}$ is based on the distance d (euclidean distance between the target point and the original point), and p is a factor that determines the effect of distance "d" on the interpolated feature. $\hat{y}$ is the result of the Nearest Neighbor Interpolation of actual neighbours y_i.

Unbalanced Dataset Problem

Unbalanced data sets occur in two class domains when the number of instances belonging to one class is significantly larger than the number of instances of the other class.

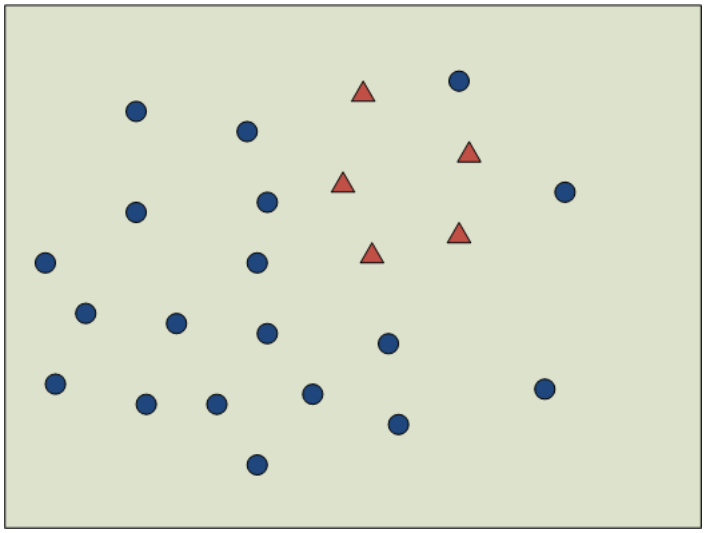


Figure 3.2: A state of an unbalanced data set problem in two class problem represented in a two-dimensional euclidean space. In this situation, the class represented by triangles is the minority class and the class represented by squares are the majority class.

Figure 3.2 depicts a two class problem in a two-dimensional euclidean space. In real life applications, it is not always possible to acquire the same number of instances for every class. This might occur due to a lack of sufficient knowledge about the minority class (the class containing only few instances), for example difficulties collecting information about rare species.

Solving the problem of unbalanced datasets is of utmost importance as it can directly affect a classifier's performance. In fact, in such scenarios, classifiers can often predict the majority class with a relatively high accuracy, yet always misclassify the minority class; although in many data mining applications, such as in medical diagnosis domains, classification of the minority class is of crucial importance. Moreover, a classifier can reach a very high overall accuracy but still perform poorly when classifying the minority class. This can be observed from the following accuracy measure:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.3}$$

where TP, TN, FP and FN correspond to the true positive, true negative, false positive and false negative values respectively. In the case of a minority class where $TN + FP \gg$

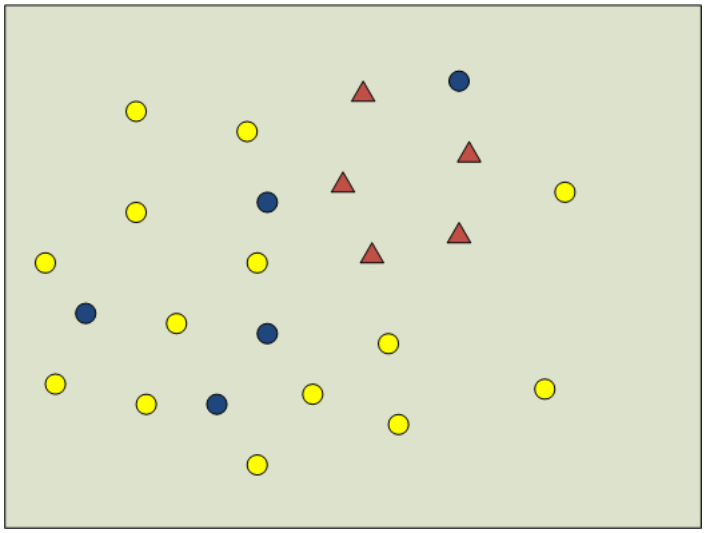


Figure 3.3: A state of balanced data set in two-class problem represented in a two-dimensional euclidean space. In this situation, the class represented by triangles is the minority class and the class represented by blue circles are the majority class. Instances represented in yellow circles are omitted by RUS to form a balanced data set

$TP + FN \geq 0$, i.e. TP and FN are relatively small values (which represent the minority class) in comparison to TN and FP which are associated with the majority class. Consequently, the corresponding accuracy measure will be misleading as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \approx \frac{TN}{FP + TN} \tag{3.4}$$

As discussed in section 2.4, Hulse *et al.* [41] performed a comprehensive experiment on different solutions for the problem of unbalanced data sets, using 35 real world benchmark data sets, and 11 learning algorithms. The authors ranked the Random majority Under-Sampling (RUS) as the sampling technique with the best results, followed by the Random minority Over-Sampling method (ROS). It was concluded that these two sampling techniques performed much better than other intelligent schemes such as SMOTE [20] or OSS [49]. Based on the aforementioned results, we utilized ROS and RUS in PROFECE.

The Random majority Under-Sampling technique (RUS) discards some instances from the majority class in order to present a balanced data set to the classifier. If

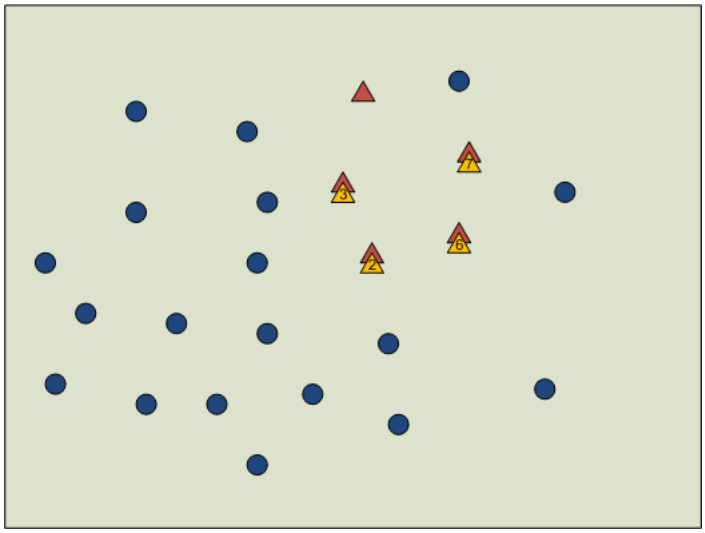


Figure 3.4: A state of a balanced data set in a two-class problem represented in a two-dimensional euclidean space. In this situation, the class represented by triangles is the former minority class and the class represented by blue circles are former majority class. ROS replicated instances from the minority class. Yellow triangles shows the number of replicas per instance

we examine the data set illustrated in figure 3.2. In order to apply RUS, there is a random selection of majority class instances (circles). The selected instances are omitted, and as a result, a state of balanced data set is reached as in figure 3.3 where the yellow circles are the selected instances to be omitted. The instances represented by blue circles and by triangles form the balanced data set.

The Random minority Over-Sampling technique (ROS) works by replicating instances from the minority class. The synthetic instances in this case are a replica of existing instances. We can assume the data set illustrated in figure 3.2 is the starting point. ROS replicates random instances of the minority class (represented by red triangles). It might create more than one replica from a single instance, as shown in figure 3.4 where yellow triangles represent the number of replicas created per instance.

In this research we applied several techniques to solve the unbalanced data set problem. More specifically, we applied the following techniques:

- Random majority Under-Sampling (RUS)
- Random minority Over-Sampling (ROS)

- Weighting Factors, which is described in section 3.5.2.
- The utilization of several fitness functions on genetic programming to solve the unbalanced data set problem, as mentioned in section 2.4.2 and its enhancements, as proposed in Section 3.4.4.

We also discuss an analysis on the effects of each technique on classification results based on haptic information.

3.3.2 Feature Selection and Generation Techniques

Learning from data or data mining can be affected by many factors. The data and the qualities it presents might be unreliable, noisy, redundant, or irrelevant. For these reasons, the learning process might degrade. The feature selection process identifies and omits the irrelevant and noisy attributes. Consequently, the process targets the capture of relevant and low noise attributes [38].

Feature selection algorithms can be categorized into wrappers and filters. Wrappers use the accuracy evaluations that would be applied in the learning process to determine what features to omit or to include. However, filters use general measurements, that are not related to the learning algorithm, to identify irrelevant and noisy attributes.[38].

Linear Forward Selection

Linear Forward Selection is based on Sequential Forward Selection (SFS) that performs a hill climbing search on attributes. The process starts with an empty set of attributes. Then, a single attribute evaluation is performed. The best score attribute is added to the subset and so on, until a relative error, defined by the algorithm, occurs. Lets assume a data set with N attributes. The number of evaluations for the former algorithm is $\sum_{i=0}^{N}(N-i) = \frac{N(N+1)}{2}$. Such quadratic growth is undesirable for high dimensional data sets. Therefore, Linear Forward Selection limits the number of features to be selected in every step with a user-specified constant [34]. It therefore decreases the number of evaluations and the complexity of the algorithm. Linear Forward Selection limits the number of features based on two methods [34]:

- Fixed Set: Prior to ranking the attributes, the algorithm selects the top k-ranked attributes as an input for forward selection. This method utilizes the top k-ranked attributes and omits the remaining attributes, thus resulting in a number of evaluations equal to $= \frac{K(K+1)}{2}$. The number of evaluations is not bounded by the data set in question. The top k-ranked attributes, considered the highly relevant features, are based on a single attribute score, which might result in attributes being omitted based on a single-attribute evaluation. Yet, those attributes might have high scores if combined with other attributes.

- Fixed Width: In this method, rather than only selecting from the top k-ranked attributes, it adds the top attributes that are not included in the top k attributes, based on the initial ranking when an attribute is selected permanently from the top k-ranked attributes. The window width is therefore fixed. It gives a chance to attributes that are not among the top k-ranked attributes to be part of the selected attributes. The number of evaluations of this method is $N \times k - \frac{k(k-1)}{2}$ where N is the number of attributes. Thus, the fixed width method incorporate attributes with lower scores, in comparison with the fixed set method.

The reason for utilizing Linear Forward Selection for PROFECE is because of its efficiency and low cost in terms of computational overhead.

Forward Selection with Subset Size Determination

Gutlein et al.[34] utilized m-folds cross validation in determining the best instance to incorporate permanently into a subset. The m-fold cross validation splits the data set into m different learning data sets where m forward selection (Section. 3.3.2) is performed. Then it utilizes the results of the test sets to form a decision, by favouring the best subset and its size. In order to favour a subset, an average score is calculated based on the m-test data sets and based on the size of the subset as well.

3.3.3 Best First Feature Selection

Best First Feature Selection is based on a greedy hill-climbing search. The algorithm is designed to search greedily either in a forward direction, where it starts with an empty feature set and adds up relevant features (forward selection), or in backward direction, where it starts with the complete set of attributes and omits irrelevant features (backward elimination). The forward selection extends the selected subset as long as there exists

a feature that decreases the relative error, while the backward elimination decreases the feature subset as long as the removal of a feature does not increase the relative error measure. Best First Feature Selection, using a greedy search, guarantees to find a local minimum but not a global minimum [74].

3.3.4 Classification Techniques

Naïve Bayes

Naive Bayes is built upon a simple and clear approach for probabilistic knowledge utilization, representation, and learning[45]. It is built to tackle inductive supervised tasks. The algorithm is "naive" because it is based on the following two assumptions[45]

- predictive variables are conditionally independent from each other, for a given class.
- no existence of attributes that are latent or hidden impacts the prediction process.

Further, Naive Bayes calculates a given class probability and the probability of an attribute vector, given the probability of a given class, as the following:

$$p(C = c \mid \mathbf{X} = \mathbf{x}) = \frac{p(C = c)p(\mathbf{X} = \mathbf{x} \mid C = c)}{p(\mathbf{X} = \mathbf{x})} \tag{3.5}$$

where

where C is a random variable representing the class of an instance, c a particular instance's class, X is a vector of random variables representing the observed attributes, and x is a single observed attribute vector value that is to be tested[45]. Since Naive Bayes assumes that attributes tare conditionally independent, thus [45]

$$p(\mathbf{X} = \mathbf{x} \mid C = c) = \prod_i p(X_i = x_i \mid C = c) \tag{3.6}$$

One common assumption in the Naive Bayes algorithm is that in each class the attribute values are normally distributed, thus [45]

$$p(\mathbf{X} = \mathbf{x} \mid C = c) = g(x; \mu_c, \sigma_c), \text{ where} \tag{3.7}$$

$$g(x; \mu_c, \sigma_c) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3.8}$$

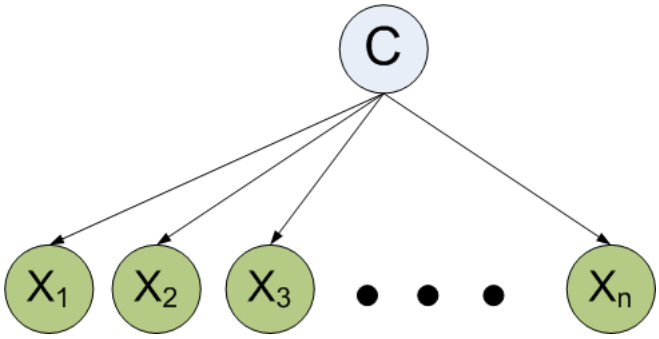


Figure 3.5: A naive bayes classifier where the attributes X1,X2,..,Xn are independent from the class attribute C[45]

As it can be observed, Naive Bayes computes a few and simple parameters from the training set, such as the mean and standard deviation, leaving no room for any external parameters to be adjusted. Although Naive Bayes is a simple classifier, its classification results are comparable to other more intelligent and complex classifiers. Thus, for its simplicity and accuracy, we decided to use Naive Bayes in our work.

k-Nearest Neighbor (KNN) and Instance Based Learning (IBL)

Instance Based Learning [5] is an enhancement to the popular K-Nearest Neighbor classifier [21]. Among many reported problems of K-Nearest Neighbor [14], Instance Based Learning overcomes the computationally expensive problem where KNN stores all the instance data and is non tolerant to noisy data. IBL is an incremental learning algorithm that outputs a concept description, where a concept is a function that assigns instances to categories[5]. The concept description contains a selected subset of training data and probably additional information about their performances (e.g. previous number of correct classifications ..etc). The subset can be updated incrementally as the training process is conducted[5]. IBL algorithms consist of the following three components[5]

- Similarity Function: a numeric value that compares the similarity between the concept description of stored instances and of a given training instance t.
- Classification Function: A component that classifies based on the result of the similarity function.
- Concept Description Updater: A component that takes care of updating the description concept by observing the result of the similarity function and of the clas-

sification performance. Moreover, it selects which instances from the concept description will be updated.

IBL is based on two assumptions [5]

- "similar instances have a similar classification"[5]. This assumption leads to a bias toward neighbour instances and thus results in the second assumption.
- Attributes have equal weight in forming the classification decision.

IBL applies the following similarity function [5]

$$sim(x; y) = \sqrt{\sum_{i=1}^{n} f(x_i, y_i)} \tag{3.9}$$

where [5]

$$f(x_i; y_i) = (x_i - y_i)^2; x_i, y_i \text{are numeric numbers.} \tag{3.10}$$

In a nutshell IBL is simple and noise resistant classification algorithms.

Random Forest

Random Forest [13] is a classifier that builds an ensemble of decision trees and utilizes those ensembles to vote for the predicted class, where a classification is based on the results of this collective decision. Thus, a class is predicted based on a majority vote. A single tree of the forest is grown as the following:[13]

1. Given a data set of N instances, select N instances with replacement as a training data set to form a tree(bagging).
2. In case the data set consists of M features, randomly select a number m such that $m \ll M$, m features are randomly selected for every tree grown in the forest. Moreover, there are no pruning for each tree grown.

The utilization of bagging enhances accuracy and makes it possible to estimate a generalization error for the ensemble of trees. This occurs since bagging selects with replacement, therefore out-of-bag data sets can be utilized by accumulating the votes from trees that didn't use an instance x to be tested. This type of estimate is unbiased and consequently eliminates the need for cross-validation, as the testing is performed internally[13].

In addition, a percent of the misclassification rate is utilized to measure a features importance. This is performed as the following: [13]

1. Assume a data set of M features. Select a feature m and permute it across the out-of-bag instances (as a noise).
2. The selection of m is incremental from 1 to M.
3. Test the modified out-of-bag instances to the constructed trees.
4. Test the original out-of-bag instances to the constructed trees.

Random Forest classification performance is affected by the strength of individual trees constructing the forest and the correlation among the trees. Therefore, as described earlier, randomness is inserted to reduce correlation among trees[13].

Random Forest potency is materialized by the following characteristics [13]

- High accuracy.
- Robustness to noise and outliers
- Internal measurements of error, strength, feature importance, and correlation.
- The process can be parallelized.

The objective of Random Forest in PROFECE is to achieve the aforementioned characteristics.

Neural Networks (NN)

Neural Networks imitates the human brain conceptually and functionally, where a human brain contains billions of neurons (estimation between 50-100 billions). Neurons are "an electrically excitable cells that processes and transmits information by electrical and chemical signalling" [3] that are interconnected to form a network, which is the main component of the nervous system. The transmission of information occurs chemically via synapses[3]. Artificial Neural Networks is designed with the target of achieving human-like performance in speech and face recognition[52]. Neural Networks form models that discover various hypotheses simultaneously, and is based on parallel connected nodes with varied weights[52]. There are several neural network models that can be divided into two general categories:[42]

- Static networks: consists of memory-less nodes, where output is independent of previous or future inputs, but depends solely on current inputs. The most common examples of static Neural Networks are MultiLayer Perceptron (MLP) Neural Network and Radial Basis Functions (RBF).

- Dynamic networks: nodes with memory, consisting of equations usually represented by differential equations such as the Time Delay Neural Network (TDNN)[42].

Static Network utilizes nonlinear transformations in such $u = G(x)$, where $x \in \Re^n$, $u \in [0,1]^m$ or $u \in \Re^m$, n and m are the dimensions of x and y respectively[42]. Before we describe MLP, we will describe what is a perceptron. The perceptron was introduced by rosenblatt[67] where an n-dimensional vector is an input and the perceptron considers a wieghted sum of the inputs, in addition to a bias θ. Moreover, it applies a nonlinear activation function on the result by using the sigmoid activation function [42]

$$f_s(y) = \left(1 + e^{\beta y}\right)^{-1} \tag{3.11}$$

where β determines the steepness of the sigmoid function. The first perceptron introduced by rosenblatt[67] uses a hard-limiter activation function, thus

$$f_{HL}(y) = \left\{ \begin{array}{l} 1 \; y > 0 \\ 0 \; y \leq 1 \end{array} \right\} \tag{3.12}$$

The perceptron that uses the sigmoid activation function represents the main component of the Multilayer Perceptron Network.

MultiLayer Perceptron Network (MLP) The usage of Single perceptron [67] limits the capability to linear decision boundaries. However, cascading the perceptrons (neurons) and connecting the output of each neuron of one layer to the inputs of all neurons of the next layer $l + 1$forms the MLP neural network. Such architectural upgrades increases the capabilities of Neural Networks so that it can find non-linear boundaries [42]. The capabilities of MLP Neural Networks can be categorised in the following three ways: [42]

- Can implement boolean logic functions.
- Non-linear partition of the pattern space.
- Non-linear transformation for functional approximation can be implemented.

As the haptic data and the handwritten signature represent a complex problem with non-linear solutions, and given the capabilities presented by MLP-NN, evaluating the performance of MLP-NN in comparison to other classifiers motivated the inclusion of such a popular classifier in PROFECE.

Support Vector Machine

Support Vector Machines (SVM) is one of the most popular classifiers currently applied. The initial development of SVM was motivated by the following segment of problems [15]

- The bias variance tradeoff [33]
- Capacity control [36]
- Overfitting [57]

The core idea of the aforementioned problems are similar. It can be clarified as the following:[15] Assume a given training data set for the purpose of a given learning task. In order to achieve an optimal generalization performance (test data's error rates), a balance should be maintained between accuracy and capacity where capacity is the ability to learn new training data with no errors. High capacity means observing a picture with fine details, and when a small detail is missing on the test data presented, the classifier will reject it. For example a picture of a coupe car is presented, and because it only has two doors it won't be considered a car since the training data only contains a car with four doors. However, little capacity means that if the object presented is a red ball, it would be perceived as a car since it matches the color of the car in the training set. Therefore, the tuning of the optimal generalization motivated the creation of SVM[15].

SVM is intended for regression, classification, and feature extraction problems. The SVM mechanism works by mapping a non-linear input space into a higher dimensional feature space, to find a separable hyperplane that maximizes the margin of separation among classes. Thus [40]

$$\Phi : R^N \rightarrow F \tag{3.13}$$

where F is the feature space which requires a dot product evaluation [40]

$$k\left(x, y\right) := \left(\Phi\left(x\right).\Phi\left(y\right)\right) \tag{3.14}$$

Once a separable linear hyperplane is found in the high dimensional space, a non-linear decision boundary is formed in the input space. After such a step, SVM utilizes the resulted non-linear decision boundary in the input space, with the use of kernals as

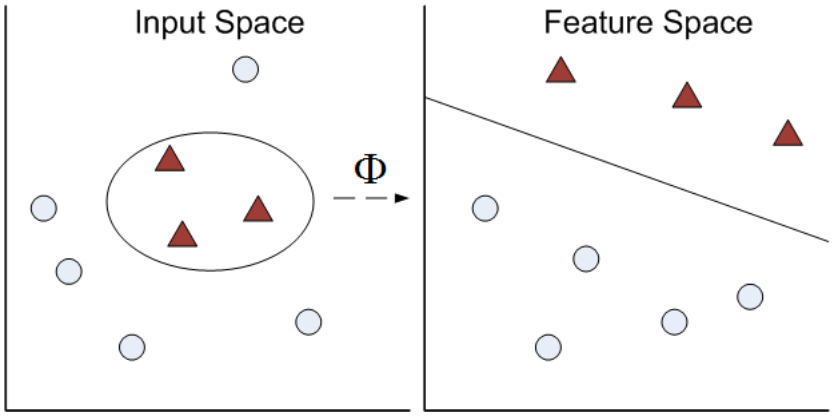


Figure 3.6: A mapping nonlinear data from a specific dimension space into a higher dimension feature space based on Φ and find a linear separable hyperplane that maximize the margin. As a result non-linear decision binary is formed in the input space and can be easily computed in the input space using kernals with out going back to the feature space[40]

a model for incoming test instances as in fig. 3.6[40]. There are several kernals used such as [40]

- Polynomial kernal, where $k(x, y) = (x.y)^d$
- Radial Basis Function(RBF) kernals for example $k(x, y) = exp\left(-\|x - y\|^2 / (2\sigma^2)\right)$
- sigmoid kernals $k(x, y) = tanh(k(x.y) + \Theta)$ where k is gain, and Θ is the offset.

The main advantage of SVM is its ability to create complex model that deals with real life problems and yet is easy to be analyzed mathematically, in contrast to Neural Network. In addition, SVM's find the proper linear hyperplane in a higher dimensional feature space, however, it utilizes the non-linear decision line in the input space. Calculation in a higher dimensional feature space is therefore not required while testing new data sets[40].

3.4 PROFECE Advancement

In this section, we will present the PROFECE building blocks that contain proposed contributions. The contributions fall into proposed methods of feature selection, based on

genetic programming, and proposed fitness functions, to guide the evolutionary process in order to solve the unbalanced data set problem. The following methods can be placed in the PROFECE framework as feature selection and classification, since GP is utilized in PROFECE for feature selection and classification.

3.4.1 Evolutionary Computing

Evolutionary Computing (EC) refers to a group of algorithms that are based on the evolution of population in order to obtain a solution to certain problems. The population that consists of individuals evolves from one generation to another, to optimize the solutions provided by the individuals and ultimately reach an acceptable solution for the problem in question. EC algorithms vary in the way the initial population is selected, in the way each individual is represented, and in the evolving process from one generation to the next. There are three popular EC algorithms, namely Genetic Programming (GP), Genetic Algorithms (GA), and Evolutionary Algorithms(EA's)[2].

3.4.2 Gene Expression Programming

Gene Expression Programming (GEP) is an enhancement of Genetic Programming and Genetic Algorithms (GA). All three techniques use populations of individuals and evolve individuals using genetic variations such as mutation and crossover, based on fitness functions. GEP combines the advantages of GP and GA techniques, where Genetic Programming (GP)[47][48] is based on non-linear individuals represented in parse trees, and GA is based on individuals of linear strings (chromosomes)[29]. However, GEP individuals in which chromosomes are represented as linear strings can be translated and expressed using the KARVA language to form a parse tree[30].
GP and GA have one common attribute, which is that both are represented in a single entity (genome and phenome). Such representation would impose the following two limitations [30]

- Simplifying the gene manipulation would result in a loss of functional complexity, as in GA's.
- Increasing the functional complexity would result in difficulty of reproduction as in GP.

The GEP representation of chromosomes and phenotypes (expression trees) allows a one-to-one translation from chromosomes to expression trees. Any gene modification will always result in a valid expression tree. Therefore, evolving genomes in GEP would not require any extra measure to insure the validity of the resulted expression trees, while validation of expression trees is required in the case of GP[30].

The evolution process of GEP [30] starts with the creation of a random or predefined initial population. A translation of the k-expressions is applied which results in an execution of the program. The resulted program is measured and evaluated according to the fitness function presented. Based on the evaluation of the fitness function, the evolving process either terminates or proceeds into a further evolving process. An elitism process or a combination of elitism and random selections can be applied to select the parent individuals which will partake in a further evolution process. Thus, replication, mutation, Insertion Sequence (IS) transposition, Root Insertion Sequence (RIS) transposition, gene transposition, one point recombination, two point recombination, and gene recombination would occur to generate next generation of individuals, with the hope of a more suitable evaluation result. The process would continue until a targeted objective function is achieved or when any other constraint is reached, such as a limit on number of generations.
A more detailed description of each GEP operator can be found in [29, 30, 10].

GEP's genome or chromosome, which might consist of one gene or more, is represented in a linear fixed length string. GEP mimics the chromosomes in real life, where a coding sequence of a gene (Open Reading Frames (ORF)) begins with a start codon. Then there are the amino acid codons, and lastly comes the termination codons. Similarity, in GEP the start codon is the beginning of a gene, and the termination codons are not always the last position of a gene. Therefore, some positions in a gene sometimes become useless to this specific chromosome, while it might be utilized during the evolution process development.

An example would be the most suitable clarification of the GEP phenotype and its corresponding genotype. Consider the following expression

$$\sqrt{a} * ((\frac{a+b}{a}) - ((a-b)+b)) \tag{3.15}$$

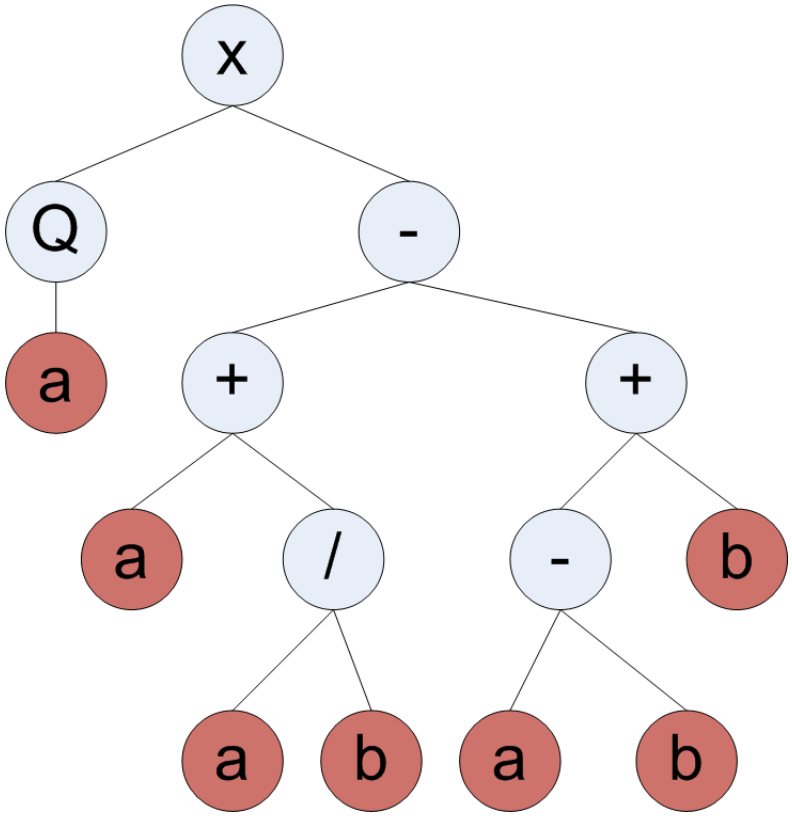


Figure 3.7: The corresponding GEP expression tree of genotype xQ-a++a/-bbaabaaabaab

which corresponds to the expression tree illustrated in fig. 3.7

where the GEP expression tree is the phenotype of the genome and the following linear string represents the corresponding genotype of the chromosome $xQ-b++a/-bbaabaaabaab$ [30], Q is the square root.

GEP genes consist of a head and a tail, where the head contains the function set elements (F) and the terminal set elements (T), and the tail contains the terminal set elements (T). In our example, the function set $F = \{x, +, -, /\}$ while the terminal set elements are $T = \{a, b\}$. One important constraint is the length of a gene tail t, which can be calculated as follows [30]

$$t = h(n-1) + 1 \tag{3.16}$$

where n represents the maximum number of arguments for any element in the function set F.

3.4.3 Proposed Further Utilization of GEP for Feature Selection and Generation

In this thesis, we utilize GEP to achieve the following

- Generation of analytical functions that will be utilized as classifiers for user identity verification.
- feature reduction where it finds a minimal knowledge preserving subset of features.

An evaluation and computation of a generated analytical function $f(v_1, \cdots, v_n)$, where v represents the variables (features), is performed by a comparison with a threshold τ for user identity verification, thus

$$h \in C_p; f_i(v_1, \cdots, v_n) \geq \tau \tag{3.17}$$

$$h \in C_n; f_i(v_1, \cdots, v_n) < \tau \tag{3.18}$$

where h is the predicted class, C_p denotes the positive class and C_n represents the negative class. In the latter case, GP-generated analytical functions are utilized in order to exploit relevant features (from the function variables) in our high dimensional (haptic) data sets. In this research, feature reduction is based on a conservative approach where a union of all variables (features) presented, in all resulted generated analytical functions of cardinality N_{GP} for relevant feature selection. Features are selected using N_{GP} independent GEP runs (from one source data set). The formalization of this process can be described as follows Let $V_p = \{v_1, v_2, \cdots, v_{N_{V_p}}\}$, $N_{V_p} = card(V_p)$ be the set of predictor variables (i.e., features, attributes) of a data set and let y be the predicted variable. Moreover, let $\mathcal{F}_{GP} = \{F_1, F_2, \cdots, F_{N_{GP}}\}$, $N_{GP} = card(\mathcal{F}_{GP})$ be a collection of GP models estimating y and derived from the same data set having V_P and y as predictor and predicted variables respectively. The models in $\mathcal{F}_{GP}$ have the form

$$\hat{y}_k = F_k(v_{k_1}, v_{k_2}, \cdots, v_{k_{p_k}}) \tag{3.19}$$

where $k \in [1, N_{GP}]$ and arity $p_k \in [1, N_{V_p}]$. The set $V_{GP}^k = \{v_{k_1}, v_{k_2}, \cdots, v_{k_{p_k}}\}$ consist of the arguments of the k-th GP model. $\hat{y}_k$ denotes the estimation of y generated by the

k-th GP model with arguments given by $v_{kj} \in V_p$ for all $k \in [1, N_{GP}]$. Define the set

$$\mathcal{R}_{V_p,\mathcal{F}_{GP}} = (V_p/\mathcal{F}_{GP}) = \left[\bigcup_{k=1}^{N_{GP}} V_{GP}^k\right] \subseteq V_p \tag{3.20}$$

and call it the *relevant subset* of V_p given $\mathcal{F}_{GP}$. It represents the (minimal) subset of predictor variables required by the collection of GP models estimating y. From it, a *relevant ratio* is defined as $\lambda_{\mathcal{R}_{V_p,\mathcal{F}_{GP}}} = card(\mathcal{R}_{V_p,\mathcal{F}_{GP}})/N_{V_p}$, as a measure of the degree of dimensionality reduction achieved by variable selection associated to $\mathcal{F}_{GP}$ models. Similarity, the set

$$\mathcal{I}_{V_p,\mathcal{F}_{GP}} = \{V_p - \mathcal{R}_{V_p,\mathcal{F}_{GP}}\} \tag{3.21}$$

is defined as the *irrelevant subset* of V_p given $\mathcal{F}_{GP}$. It is the set of variables that is absent in any of the $\mathcal{F}_{GP}$ and therefore, superfluous from the point of view of the GP models. An *irrelevant ratio* is defined as

$$\lambda_{\mathcal{I}_{V_p,\mathcal{F}_{GP}}} = 1 - \lambda_{\mathcal{R}_{V_p,\mathcal{F}_{GP}}} \tag{3.22}$$

Moreover, an alternative approach is proposed for the exploitation and further analysis of GP-based feature selection. It is based on the following

$$R_q^\alpha \subseteq \mathcal{R}_{V_p,\mathcal{F}_{GP}} \tag{3.23}$$

where R_q^α denotes the subset of the most frequent variables, understood as those contained within the α-percentile of the cumulative frequency distribution. Specifically, this refers to the cumulative frequency distribution of all variables exploited in N_{GP} models that are generated using N_{GP} independent GEP runs. These two feature selection methods are applied as a mean of dimensionality reduction on the virtual check application, as will be discussed in the next chapter.

3.4.4 Proposed Solution for the Unbalanced Data Set Problem from a Gene Expression Programming Perspective

In order to solve the unbalanced data set problem from a GEP perspective, and after a second look into the evolutionary process in genetic programming, one way to solve the unbalanced data set problem is to consider specially designed fitness functions to guide the evolutionary process towards analytical functions in which the unbalanced data set problem is avoided. The evaluation process is performed in every iteration in

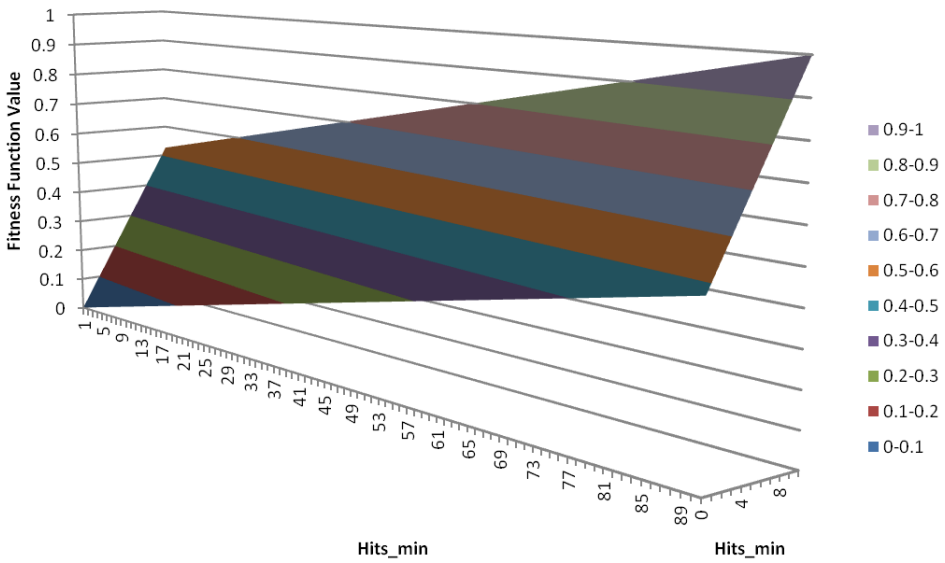


Figure 3.8: The behavior of the proposed New Fit1 function based on an unbalanced data set ratio of 90:10 for majority class and minority class

genetic programming. Moreover, the decision of which individual to select and when to terminate the evolutionary process is influenced by the evaluation of a fitness function. As a solution to the unbalanced data set problem we propose three fitness functions, with the intention of guiding the evolutionary process to analytical functions that mitigate the unbalanced data set problem. The three proposed fitness functions are:

$$\text{New Fit1} \ = \frac{\left(\frac{N_{min}}{N_{max}} \times hits_{max}\right) + \left(\left(1 - \frac{N_{min}}{N_{max}}\right) \times hits_{min}\right)}{2 \times N_{min} - \frac{N_{min}^2}{N_{max}}} \tag{3.24}$$

where N_{min} and N_{max} are the number of instances for the minority class and the overall number of instances, respectively. $hits_{max}$ denotes the overall correct classification and $hits_{min}$ is the correct minority class prediction. The fitness functions behaviour is illustrated in fig. 3.8

The second fitness function proposed is a non-linear function that considers the overall accuracy, in addition to the minority class and the majority class.

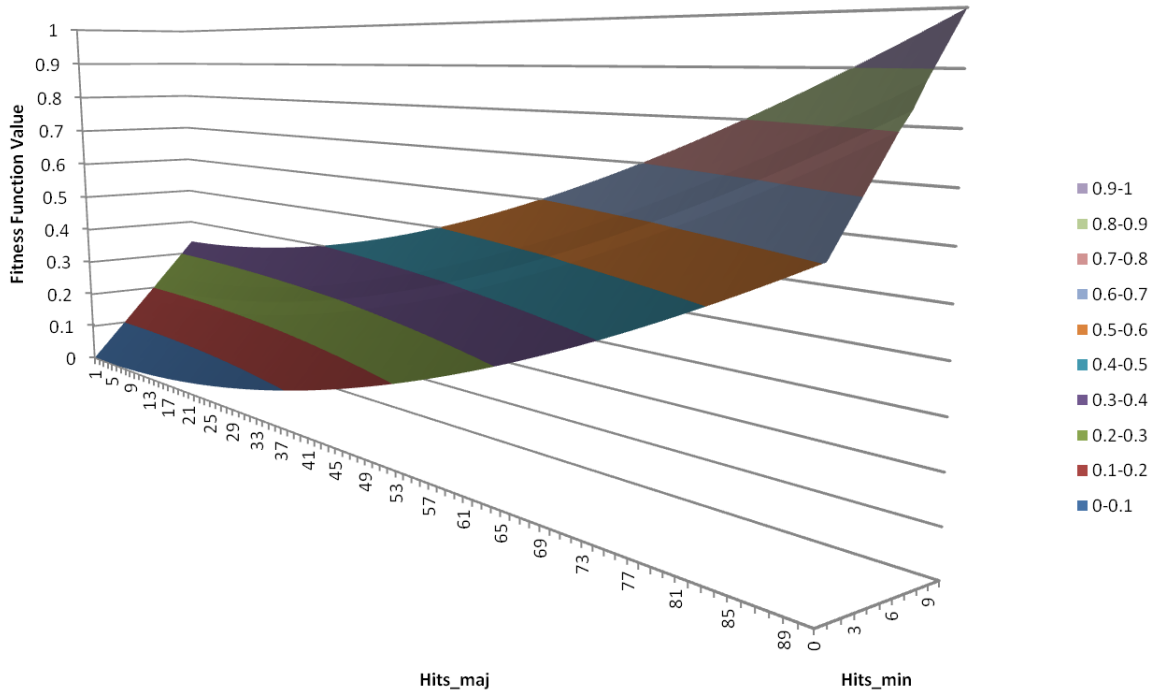


Figure 3.9: The behavior of the proposed New Fit2 function based on an unbalanced data set ratio of 90:10 for majority class and minority class

$$\text{New Fit2} \ = \frac{\frac{hits_{min}}{N_{min}} + \left(\frac{hits_{maj}}{N_{maj}}\right)^2 + \frac{hits_{max}}{N_{max}}}{3} \tag{3.25}$$

The behavior of equ. 3.25 is illustrated on fig. 3.9

Finally, the third proposed fitness function is a non-linear function that considers both the minority class and the majority class.

$$\text{New Fit3} \ = \frac{\frac{1}{1+e^{\left(\frac{Hits_{min}}{N_{min}}\right)^2}} + \frac{1}{1+e^{\frac{Hits_{maj}}{N_{maj}}}} - 1}{\frac{2}{1+e} - 1} \tag{3.26}$$

The fitness function behavior is illustrated on fig. 3.10

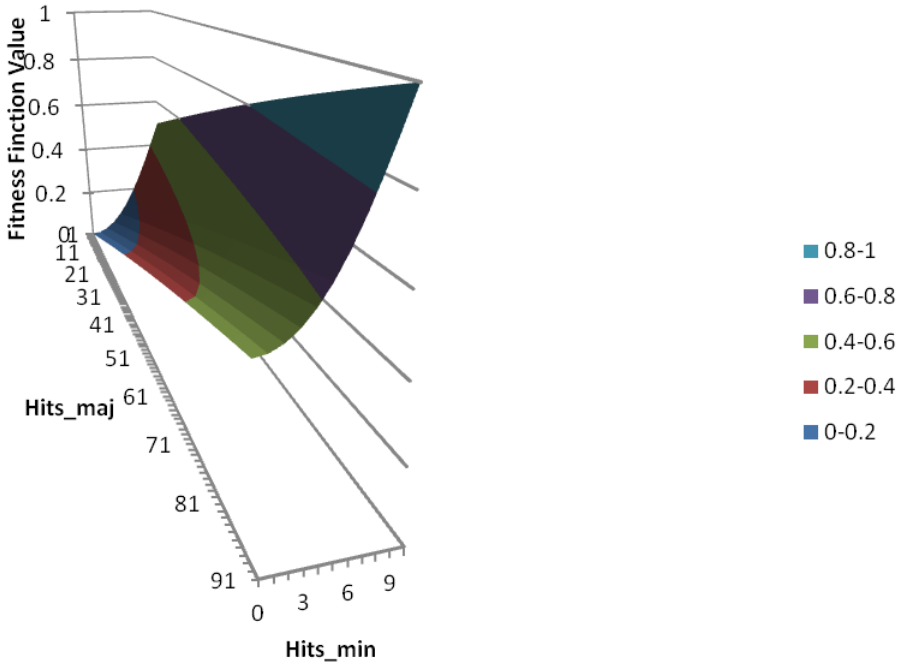


Figure 3.10: The behavior of the proposed New Fit3 function based on an unbalanced data set ratio of 90:10 for majority class and minority class

3.5 User Authentication

As many methodologies have been introduced and proposed in this research, we will try to provide additional explanations in addition to section 3.1 to describe the mixture of classifiers and feature selection methods that has been utilized in this work.

Since the instantaneous state of the haptic system is characterized by a number of haptic data types including three-dimensional position, force, velocity, etc., and since the haptic information is recorded in a frequency of 1kHz, a large quantity of haptic information is recorded. The representation of instances and the process of user authentication in this work is addressed by two different approaches.

3.5.1 Approach 1: Haptic Feature Dimensionality Reduction

The first approach, represented by a large feature vector that is formed by combining all instances of the haptic features into a single instance, with the intention of Feature Di-

mensionality Reduction. As data provided is time-varying, an instance in this approach is the number of haptic data types (position, force, etc.) times the number of samples acquired, per data type, during each acquisition process. This leads to significantly large haptic feature vectors. In this approach, in order to have a common length among instances, the feature vectors are normalized using Nearest Neighbor Interpolation, as discussed in section 3.3.1. As the resulted feature space is high (e.g. 10,000 features), this approach allows for the discovery of various feature selection methods such as the classical attribute selection method (Section 3.3.2 and the utilization of evolutionary algorithms, specifically GP (Section 3.4.1), to generate analytical functions for classification and feature selection. The utilization of genetic programming for feature selection in this approach is performed in several ways. First it utilizes the resulted analytical functions, which use a subset of the feature space. Moreover, we proposed two different feature reduction approaches (Section 3.4.3) that use the resulted analytical functions and extract a feature subset that relies on a group of individuals. In this approach, several techniques such as ROS and RUS (Section 3.3.1), in addition to the proposed GP fitness functions (Section 3.4.4), have also been utilized to solve the unbalanced data set problem. These guide the evolutionary process towards analytical functions in which the unbalanced data set problem is mitigated.

3.5.2 Approach 2: Time Varying Approach

The second approach is represented by a small feature vector, where every single set of haptic features forms an instance. To achieve better generalization [42] , this approach is more suited for classifiers that perform better when exceeding a minimum number of training instances such as MLP-Neural Networks (Section 3.3.4). The use of feature selection methodologies are sometimes limited due to the small size of feature space, however classification is still applicable. In this approach, we perform a filtering process that generates the mean values (μ), and the variance values (σ^2) of each feature, as described in [7], every 10 milliseconds. Moreover, we propose the following method where a classifiers (MLP NN, Random Forest, Naive Bayes) outputs form a set of user probabilities for a legitimate user,$O = \{p(u_i|t, L)\}$ where $p(u_i|t, L)$ is the probability of user u_i at trial t, given that we have a base set of haptic interactions L in the learning part. We assume

$$\sum_{i=1}^{n} p(u_i|t, L) = 1 \tag{3.27}$$

and

$$0 \leq p(u_i|t, L) \leq 1 \tag{3.28}$$

where n is the number of users who will be identified. The user identification is performed by selecting the user with the maximum value of $p(u_i|t, L)$ [8].

If the unbalanced data set problem exists, it is resolved in this approach by using weighting factors. Minority classes are assigned with higher weights, while lower weights are assigned to majority classes, to balance the effect of each class on the classifier by using the following equation.

$$p(u_i|t, L) \times w_i \leq w_i \tag{3.29}$$

Where

$$w_i = f(n, \sum_{j=1}^{n} s_j, s_i) \tag{3.30}$$

where the weight w_i is determined by the number of classes n and the number of instances s for user u. The subscripts i and j indicate the class ID of concern. We applied equation 3.31 below to determine the threshold for a given class i.

$$f(n, \sum_{j=1}^{n} s_j, s_i) = \left(\frac{\sum_{j=1}^{n} s_j}{n \times s_i} \right)^x \Bigg/ \left(c + \sqrt{\frac{\sum_{j=1}^{n} s_j}{n \times s_i}} \right) \tag{3.31}$$

where c and x are constants equal to 1.0 and 1.6 respectively (optimal values are chosen by experimenting various values) [8].

3.6 Testing and Evaluation of Methods

Evaluation is an important step in distinguishing the performance among several classifiers. In order for an evaluation to hold, the comparison should follow a systematic approach, applied on all classifiers. One main aspect of evaluation is how the testing is performed and how to make use of the available data set for performance prediction. As a starting point, the performance in our work is based on the success rate, which is the number of correct predictions. This is easily calculated through error rates, which are the number of incorrect instances obtained while predicting instances. Moreover, predicting

instances (testing) is performed mostly based on Hold-out Test Data.

Hold-out Test Data is a testing method that relies on the concept that testing on the training set is biased. Therefore, the training process uses a separate training data set. After the training process ends, a separate (hold-out) testing data set is utilized to evaluate the hypothesis. In order for the Hold-out Test Data to be representative of all experiments, the Subject Stratification process is performed, leading to a stratified hold-out.

3.7 Concluding Remarks

The proposed framework for haptic-based user authentication consist of several steps. It consists of pre-processing, feature reduction, classification, post-processing and evaluation. The benefits of utilizing each algorithm or set of algorithms vary. Algorithm 1 (Section 3.5.1) is intended to generate analytical functions that can be used as classifiers and for dimensionality reduction purposes. Moreover, as the analytical functions expose the features utilized in every model, it allows for further examination of the resulted feature subset to measure the contribution of haptic features towards identity verification. Moreover, algorithm 2 (Section 3.5.2) is developed with the intention of classification, without the use of feature reduction.

Chapter 4

PROFECE Verification with predefined tasks: Haptic-enabled Virtual Check Experiment

In this chapter, we will present the first stage of our planned experiments (Fig. 1.1). As we advance from one experiment to another, the complexity of the problem escalates in terms of freedom of movements. The first experiment is a virtual check application, where users are required to mimic the legitimate signature in order to be authenticated.

The goal of the virtual check experiment is to enhance the identity verification based on haptic information, and to analyze the contribution of each haptic feature towards identity verification. Users are required to mimic the legitimate user signature. This experiment cannot be utilized for continuous authentication, but it is the first step in discovering identity verification based on haptic information.

4.1 Design and Architecture

The identity verification process for handwritten signatures with haptic information relies on the haptic feature dimensionality reduction approach (Section. 3.5.1). Handwritten signatures can be seen as a non-continuous repetitive task that spans over a short time period. Therefore, the aforementioned approach suits this task since it allows the utilization of dimensionality reduction. The high frequency and richness of haptic information

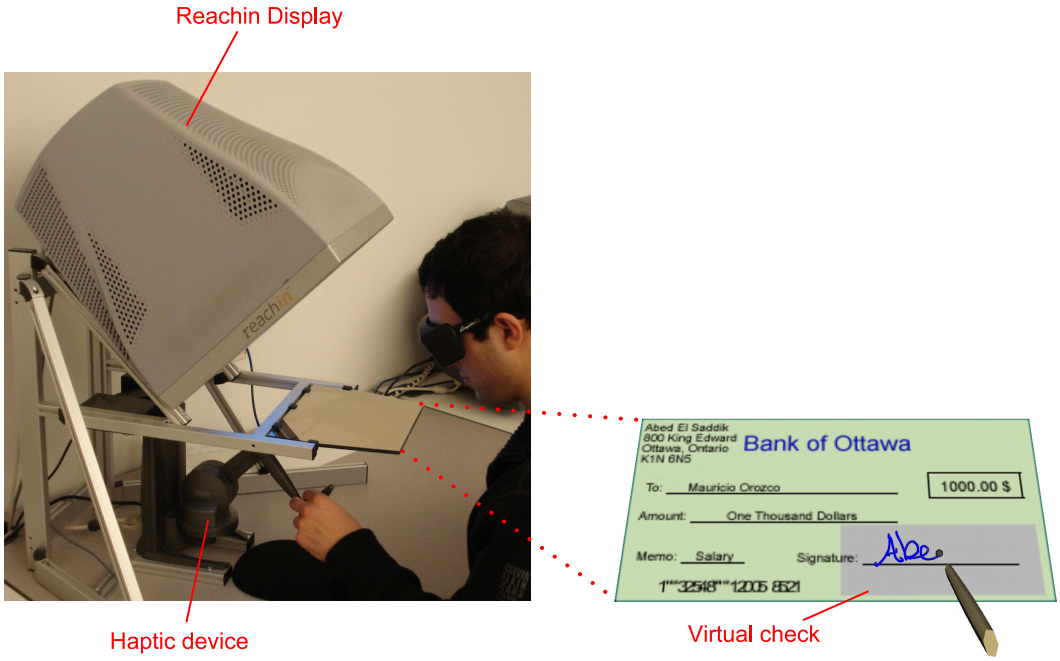


Figure 4.1: Haptic-enabled virtual check application.

results in a high dimensional vector space. In this section, we describe the design and architecture of the proposed solution for the identity verification of handwritten signatures with haptic information problem.

The overall architecture of the haptic-enabled virtual check application is shown in Fig. 4.2. The user employs the haptic-enabled virtual cheque application to sign a virtual cheque, and the observer component observes every detail of the users handwritten signature, including haptic features (such as three-dimensional position, force ..etc) and passes it to the verifier. The verifier component analyzes the captured handwritten signature and tries to match it to the legitimate user's signature, to either grant or decline access to the protected system.

Figure 4.3 shows a simple state chart diagram that summarizes, in three steps, the identity verification process of handwritten signatures with haptic information. It starts by requesting the username. Then the user is asked to sign the virtual check, and the system either grants access or requests that the user repeat the process because of a mistake in the provided username or the recorded signature. The "verifying identity" state is expanded in Fig. 4.4. This expansion shows several steps and several branches to

solve the identity verification based on handwritten signatures with haptic information. After the user provides the handwritten signature, the verification process performs the KNN interpolation described in Section. 3.3.1 where the interpolation will either up-sample or down-sample the feature space in order to reach an equal size feature space among all vectors. Moreover, a vectorization process will be applied afterwards. The data set is then divided into either a positive class or a negative class, depending on the user in question. As we follow a hold-out test data, in order to obtain a testing set that is independent from the training data set, we apply a division of the data sets into training and testing data sets. Subsequently, the observed state will depend on the selection. Whether we apply a solution for unbalanced data set in the prepossessing phase such as RUS or ROS, whether we use the original data set (IMB), whether the classifier we use is one of the classical classifiers or whether we use evolutionary computing, represented by GEP. Moreover, when GEP is applied, there are additional choices to make regarding fitness functions to be utilized, such as if the default objective function is used or if it is one of the proposed fitness functions (Section. 3.4.4. We must also decide if we will use a classical feature selection method or one of our proposed feature selection algorithms, as presented in Section. 3.4.3. The extensive state chart diagram gives a full view of the utilized algorithms and branches that can be used as a guide through the results presented in the following sections of this chapter.

The class diagram of the proposed solution to solve the problem of identity verification based on handwritten signatures is presented in Fig. 4.5. The observer component observes the users handwritten signature, including the visual and haptic information. The verifier component bases its decision on several other components, including the configuration requested, whether to consider an unbalanced data set solution or to use a certain classifier, or whether GEP will be utilized or not. The normalizer applies the KNN interpolation and is responsible of the vectorization of feature space. The resampler component is responsible of whether or not an under-sampling of the majority class or an over-sampling of the minority class is required. The GEP Feature Selection Manager analyzes the GEP generated models and utilizes them following the proposed GEP feature selection algorithms (Section. 3.4.3). The evaluator evaluates the classifier on concerns and reports the classification results.

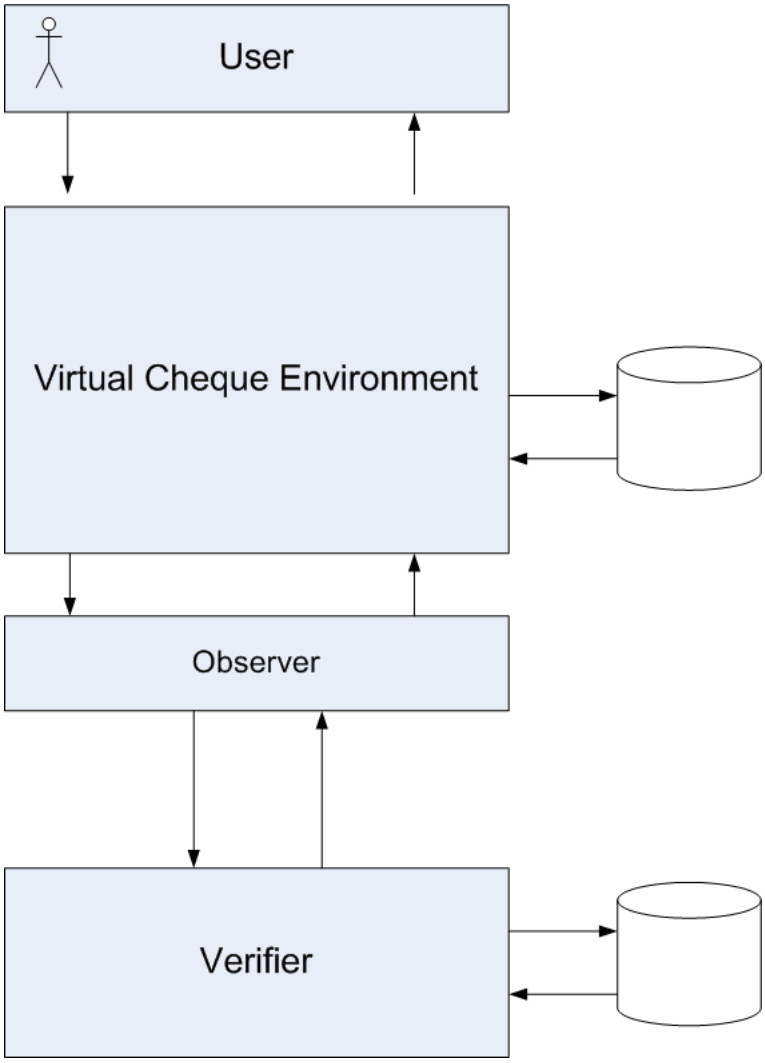


Figure 4.2: Overall architecture of Haptic-enabled virtual check application.

4.2 Identity Verification of Users based on Handwritten Signature with Haptic Information

In this section, the haptic-enabled virtual check application and the haptic data acquisition, as well as the pre-processing steps, will be described. Moreover, the utilized GEP algorithm settings will be presented.

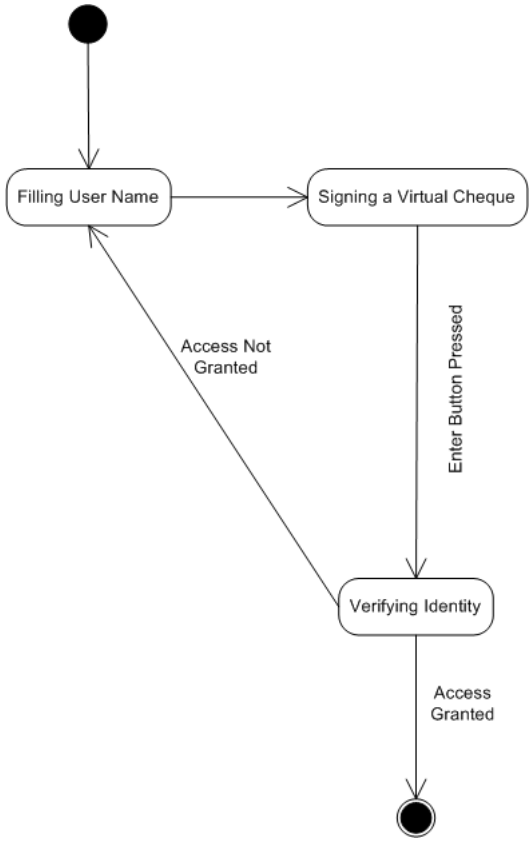


Figure 4.3: Simple State chart diagram of user verification using Haptic-enabled virtual check application.

4.2.1 Apparatus

The experiments are conducted using the Reachin Display [4] which provides stereo graphics with the addition of a haptic device, for an immersive and high quality 3D experience. The reachin visuo-haptic interface allows users to see and feel virtual objects at the same location. Users sense the haptic stimulus through the SensAble PHANTOM Desktop force-feedback device, which is equipped with encoder styles that provide a 6-degree-of-freedom single contact point interaction and positional sensing.

Moreover, as depicted in Fig. 4.1 the visual stimuli is represented by a virtual check and a virtual pen, where users can record their handwritten signature. On the contrary, the haptic stimuli represented by force and frictional feedback tries to mimic the tactile sensations felt by a user while signing a traditional paper check. The check is built on an elastic membrane surface with specific textile features, providing a realistic sensation

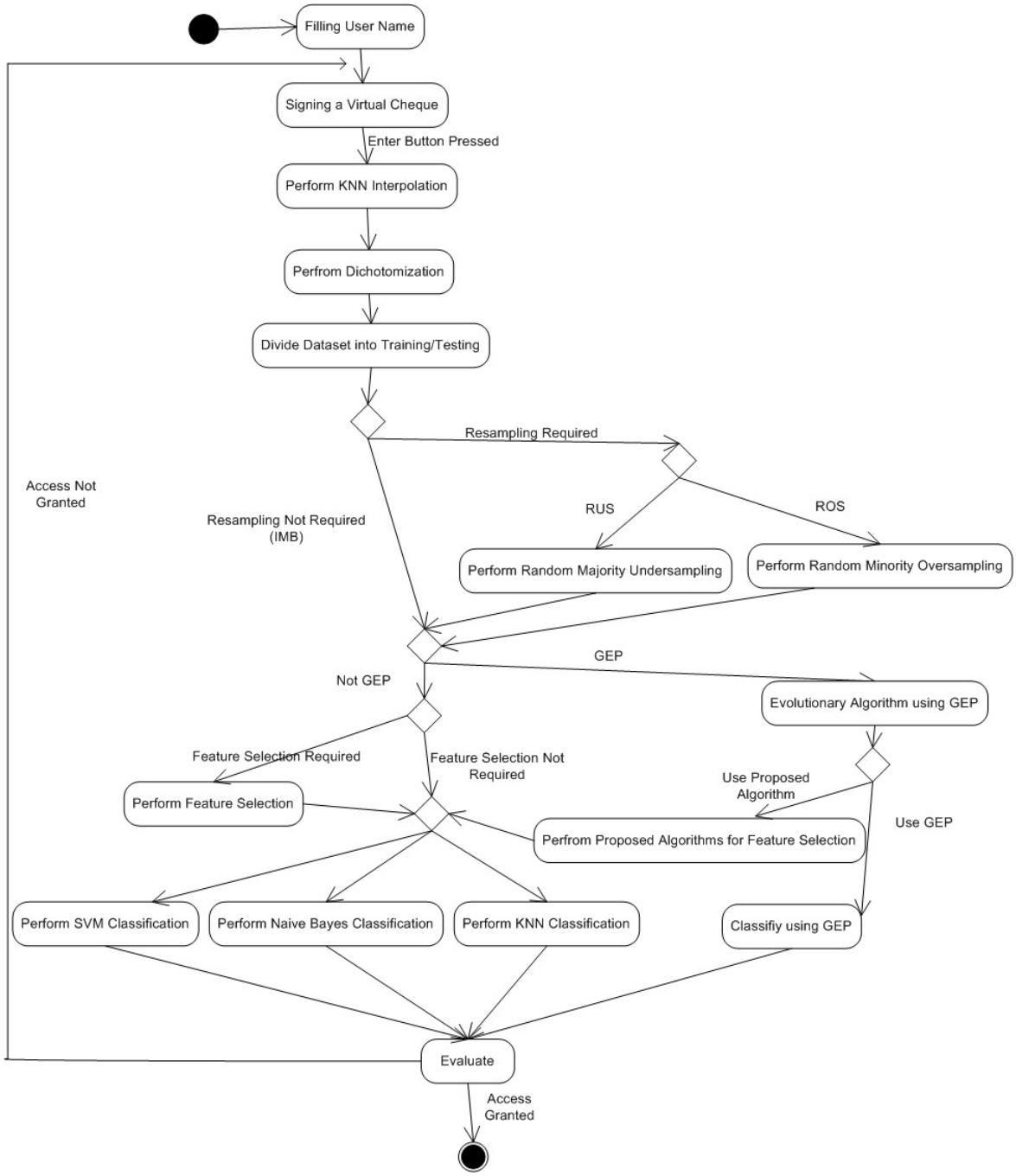


Figure 4.4: Extended State chart diagram of user verification using Haptic-enabled virtual check application.

when touching the virtual object. In addition, as with conventional dynamic signature verification technologies, the virtual check application records many attributes that re-

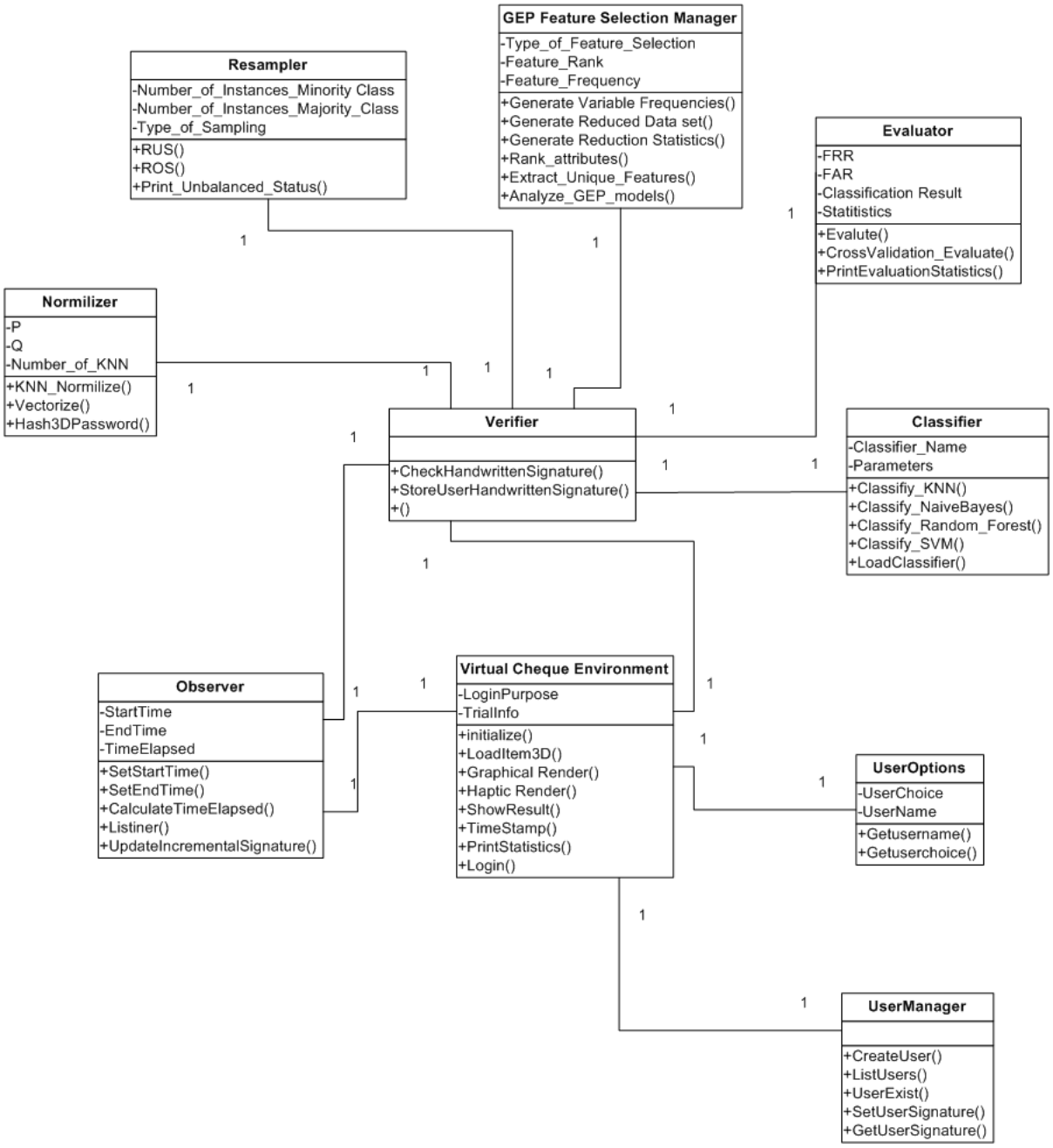


Figure 4.5: Class diagram of Identity verification using Haptic-enabled virtual check application.

flects the user's physical and behavioural traits [10].

Table 4.1: Experimental settings of the GEP algorithm.

GEP Parameter	Experimental Values
No. generations	100000
Population size	1000
No. chromosomes / individual	1
No. genes / chromosome	3
Gene head size	8
Linking function	addition
No. constants / gene	2
Bounded range of Constants	[0,10]
Inversion rate	0.1
Mutation rate	0.044
is-transposition rate	0.1
ris-transposition rate	0.1
One-point recombination rate	0.3
Two-point recombination rate	0.3
Gene recombination rate	0.1
Gene transposition rate	0.1
rnc-mutation rate	0.01
dc-mutation rate	0.044
dc-inversion rate	0.1
Function set (e.g., {function$_1$(weight), function$_2$(weight), ...})	{add(1), sub(1), mult(1), div(1)}

4.2.2 Haptic Dataset (HS1)

The haptic-based handwritten signatures are diligently gathered from thirteen different participants, with a total of ten signatures collected per individual. The data collected consists of various distinct haptic features as a function of time. The instantaneous state of the haptic system is characterized by a number of haptic data types, including three-dimensional position, force (pressure exerted on the virtual check), torque, and angular orientation. Moreover, haptic information is recorded at 100 Hz. As the data provided is time-varying, the number of attributes per signature provided is in fact the number of haptic data types (position, force, etc.) times the number of samples acquired per data type, during each signature acquisition. Consequently, it leads to

significantly large feature vectors that encompass thousands of haptic-based attributes. The resulted feature vectors were normalized to a common length of 10000 using Nearest Neighbor Interpolation, as discussed in section 3.3.1. The collected haptic data types are re-sampled(upsampled/downsampled) if required, to ensure a common feature vector length in the haptic data set. As a result, the haptic data set consists of 130 instances, where each instance is represented by 10000 features.[10]

Identity verification is a dichotomization problem (two-class identification problem). In order to examine identity verification in the aforementioned haptic data set, a dichotomizer labels instances as either A (Accept identity claim) or as R (Reject identity claim). Therefore, the pre-processing step rearranges the haptic data set into thirteen distinct data sets (each class is represented by one data set). In each set, only instances representing the positive class are marked with *accept* ($A = 1$), whereas the negative class, consisting of the rest of the data, is marked with *reject* ($B = 0$). Furthermore, the resulted thirteen data sets are further divided into 60% training and hold-out testing data of 40%, and into 80% training and hold-out testing data of 20%. Let's refer to the former data sets as *IMB_60*, and the latter data sets as *IMB_80*.

The resulted identity verification data sets were further pre-processed by applying the Random Under-Sampling (RUS) the majority class and the Random Over-Sampling (ROS) the minority class techniques, as discussed in Section 3.3.1. By using RUS and ROS, the pre-processing step obtains a balanced data set. As with the unbalanced data set, the resulted data sets (ROS and RUS) are further divided into 60% training and hold-out testing data of 40% and into 80% training and hold-out testing data of 20%. Let's refer to the former data sets as *RUS_60* and *ROS_60*, and the latter data sets as *RUS_80* and *ROS_80* [10].

4.2.3 Gene Expression Programming Algorithm Settings

The GEP experiments in this research were performed using an implementation developed at the National Research Council of Canada [73] Institute for Information Technology, using the ECJ environment [53]. In addition, we integrated the proposed fitness functions in Section 3.4.4 to the aforementioned implementation with the cooperation of National Research Council of Canada [73]. Moreover, the GEP parameter-values utilized

in the experiments are listed in Table 4.1.

The mathematical expressions in each individual are composed of one chromosome, where each chromosome consists of three gene expressions, all linked by the addition operator. The gene expressions can be formed using a combination of constants, the independent variables associated with each analytical function, and any of the functions listed. Function weightings give the user the freedom of choice in case the user prefers some operators over others. In our experiment, all operators (addition, subtraction, multiplication, and division) were assigned equal weight values (refer to Section 3.4.1 for a concise description of the GEP algorithm and its parameters). For every experiment presented, random seeds are used on each of the generated 100 runs. We would like to emphasize that for every single run, there are in fact 1000 different functions obtained (which correspond to a population of size 1000). This will be discussed further in Section 4.3

The GEP algorithm can be summarized in the following steps: (Alg.1)[29]

Algorithm 1 GEP Algorithm in general

1: Create chromosomes of initial population
2: Express chromosomes
3: Execute each program
4: Evaluate fitness
5: **if** terminate = true **then**
6: End
7: **else**
8: Retain best programs
9: Select programs
10: Perform evolutionary operations such as replication, mutation, IS transposition ..etc.
11: Prepare new programs of next generation
12: **end if**
13: Go to Line 2.

4.3 Virtual Check Experiment Results

A series of experiments have been conducted to examine the proposed genetic programming approach and to show the possible advantages of utilizing genetic programming techniques in a haptic-based identity verification system.

4.3.1 Haptic-based Identity Verification using Classical Classifiers

This first experiment is used as a foundation for comparison between classical classifiers and genetic programming techniques, when applied to the haptic-based identity verification problem. Specifically, this experiment is conducted in order to observe the binary-classification behaviour of classical classifiers (described in Section 3.3.4) when considering unbalanced, random under-sampled and random over-sampled versions of the haptic-based signatures data set. The evaluation process of classical classifiers was performed using the WEKA environment [74], and the LIBSVM [19] tool. Moreover, fine-tuning of the parameters for each reference classifier was conducted. Table 4.2 presents the verification success rate (VR), the False Acceptance Rate (FAR), and the False Rejection Rate (FRR) results using SVM, Naïve Bayes, k-NN, and Random Forest classifiers on the aforementioned data sets (IMB_60, RUS_60, ROS_60, IMB_80, etc.). (FAR represents the rate of accepting a given user x falsely as the legitimate user y, while FRR rejects a legitimate user by assuming he is an illegitimate user. The lower the FAR and the FRR are, the better the system performance is in terms of verifying usability and the authenticity of legitimate users).

Pre-processing the data by considering RUS and ROS techniques have shown positive and negative outcomes, depending on the data set on consideration and the selected classifier. For instance, when the SVM classifier is considered, it resulted in an improvement in FRR, at the expense of a decrease in the VR and the FAR. This is an encouraging result as it partially solves the unbalanced data set problem. A very noticeable observation is that, in all cases, the use of RUS data sets results in a decrease of the VR. This observation is possibly due to the significantly reduced number of training instances as a result of applying the RUS method. Conversely, in most cases, when RUS is applied, there is an enhancement of FAR on both Random Forest and Naive Bayes classifiers.

Therefore, RUS is potentially a useful technique when combined with Random Forest or Naive Bayes classifiers. Furthermore, it can be observed that despite its simple nature, k-NN achieved classification results that are nearly equivalent or superior to SVM classifiers. However, the use of k-NN shows no significant improvements in the results when balanced RUS and ROS (Mann and Whitney test, p¿0.05) data sets are considered [10].

4.3.2 Haptic-based Identify Verification using Classical Attribute Selection method

In this experiment, a recent classical attribute selection method 3.3.2 is utilized as a reference point for comparison between feature selection using a classical machine learning approach and genetic programming. The *subset size forward selection* method was performed using the WEKA environment [74]. The resulted reduced feature subsets were fed to the reference classifiers. The classification results are shown in Table 4.3 when HS1 is considered. In Table 4.3 the subset size forward selection method produced a relatively high feature dimensionality reduction, in comparison to the original 10000 attributes, while maintaining comparable classification performance results as well [10].

4.3.3 Haptic-based Identify Verification using Genetic Programming

In this section, a series of experiments were conducted to investigate the feasibility of utilizing genetic programming techniques in the haptic-based handwritten signature verification problem. For every subset of the 13 subsets in *IMB_60*, 100 independent GEP runs were generated. (i.e. 100 different analytical functions are generated for the 1st subset, 100 different analytical functions are generated for the 2nd subset and so on.) We have to emphasize that during each run, there are in fact 1000 different analytical functions that are maintained (since the GEP population size is initiated with 1000). However, an elitism process chooses only the GEP model with the best discrimination performance (based on the training data sets). This procedure is similarly applied to the other five generated data sets: *IMB_80*, *RUS_60*, *RUS_80*, *ROS_60*, and *ROS_80*. Further analysis of the results is discussed in the following subsections [10].

Table 4.2: Average Verification Success Rate (VR), False Acceptance Rate (FAR), False Rejection Rate (FRR) for unbalanced, RUS and ROS datasets, while considering all 13 classes (Section. 4). The results are obtained with parameter-fine-tuning of each reference classifier.

Rate of Training Samples	***Data-sets***	***SVM***			***k-NN***			***Naïve Bayes***			***Random Forest***		
		VR (%)	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***
60%	IMB	97.30	0.32	30.80	96.30	28.85	1.60	97.19	19.23	1.44	96.45	46.15	0.00
	RUS	89.20	10.40	15.40	89.79	15.38	9.78	75.44	5.77	26.12	88.17	13.46	11.70
	ROS	92.50	7.21	11.50	94.82	23.08	3.69	96.30	19.23	2.40	97.63	30.77	0.00
80%	IMB	98.50	0.32	15.40	94.08	19.23	4.81	97.93	11.54	1.28	98.22	23.08	0.00
	RUS	87.30	12.50	15.40	89.94	26.92	8.65	83.14	11.54	17.31	84.32	7.69	16.35
	ROS	93.80	6.09	7.69	94.67	30.77	3.21	96.15	19.23	2.56	97.34	34.62	0.00

Classification

Table 4.4, presents the VR, FAR, and FRR results when using genetic programming-generated models on all data sets (IMB_60, RUS_60, ROS_60, IMB_80, etc.). Moreover The VR, FAR, and FRR results using genetic programming models evolved with various fitness functions (NH, equation 2.2(BH1),equation 2.3(BH2), Equation 2.7(PA1), Equation 2.8(PA2), equation 3.26(EXP),equation 3.25 (SNL),equation 3.24(SL)) are presented in Table 4.8. Specifically, the classification results are based on the average performance of all generated analytical functions, for example 100 analytical functions per subset (class). On the other hand, in order to explore the GP individuals that demonstrate the highest performance, the first quartile of the best GP models (per subset) are selected. GP model selection criteria is based on each model's classification accuracy of test data. The average classification performance of the first quartile of the top GP models is depicted in Table 4.5 and Table 4.9.

Several observations can be made from the results depicted in Tables 4.4 and 4.5 and 4.9. First, RUS data sets demonstrated a poor classification accuracy, similar to classical classifiers. In addition, we can observed a high dimensionality reduction where only a small number of necessary variables and operations are required. This observation suggests that the GP-generated analytical functions are potentially low in complexity. It is important to clarify that the number of variables refers to the number of distinct variables used in the explicit equations (duplication of identical variables are not counted).

Table 4.3: Classification results of reference classifiers after applying Subset Size Forward Selection algorithm using the 13 Users data set (Section. 4). Values highlighted in bold denote equal or better results in comparison with Table 4.2.

Rate of Training Samples	***Data-sets***	***Average No. of Attributes***	***SVM***			***k-NN***			***Naïve Bayes***			***Random Forest***		
			VR (%)	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***
60%	IMB	149.5	95.86	0.48	48.08	95.71	**0.32**	51.92	**97.34**	**0.48**	28.85	95.71	**0.96**	44.23
	RUS	777.1	82.84	17.31	15.50	**88.17**	**10.90**	23.08	**79.14**	21.79	**9.62**	77.07	30.62	19.23
	ROS	243.1	**96.60**	0.96	32.75	**94.82**	**0.00**	67.31	95.27	**4.01**	13.46	95.86	**1.12**	40.38
80%	IMB	133.3	96.75	0.33	38.50	96.75	**0.00**	42.31	97.34	**0.32**	30.77	95.86	**0.00**	53.85
	RUS	547.8	80.46	**19.87**	15.50	88.46	**10.90**	19.23	81.66	18.27	19.23	79.29	21.15	15.38
	ROS	219.6	**97.34**	**0.96**	23.05	97.04	**0.00**	38.46	96.15	**3.21**	11.54	95.86	**0.64**	46.15

Table 4.4: Classification results based on the average performance of all GP-generated analytical functions (100 analytical functions per class) based on virtual check experiment (Section. 4 data set (HS1)).

Datasets	***60%***					***80%***				
	VR (%)	***FAR (%)***	***FRR (%)***	***No. of Variables***	***No. of Operations***	***VR (%)***	***FAR (%)***	***FRR (%)***	***No. of Variables***	***No. of Operations***
IMB	91.49	5.65	42.85	7.67	22.63	93.88	4.04	31.15	8.41	18.36
RUS	58.29	42.36	33.89	17.61	94.05	60.08	40.72	30.35	15.75	90.09
ROS	89.97	8.14	32.60	6.42	9.28	91.43	6.48	33.58	7.04	9.13

Table 4.5: Classification results based on the average performance of the first quartile of the best GP-generated analytical functions (per class) for 13 Users (HS1)(Section. 4).

Datasets	***60%***					***80%***				
	VR (%)	***FAR (%)***	***FRR (%)***	***No. of Variables***	***No. of Operations***	***VR (%)***	***FAR (%)***	***FRR (%)***	***No. of Variables***	***No. of Operations***
IMB	95.90	1.93	30.13	7.39	22.88	98.32	0.83	11.85	7.89	17.10
RUS	69.70	30.50	27.86	17.01	94.05	73.75	26.49	23.33	15.01	100.22
ROS	94.87	3.56	24.00	6.18	8.79	96.56	1.86	22.37	6.68	8.99

Table 4.6: Classification results while considering the union of all the features that appear in the generated GP models (per class) for 13 Users (Section. 4). Values highlighted in bold denote equal or better results in comparison with Table 4.2.

Rate of Training Samples	***Data-sets***	***Average No. of Attributes***	***SVM***			***k-NN***			***Naïve Bayes***			***Random Forest***		
			VR (%)	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***
60%	IMB	692.9	**97.50**	**0.16**	**30.77**	**97.78**	**0.16**	26.92	96.89	**0.96**	28.85	96.60	0.16	42.31
	RUS	1393.9	**93.19**	**5.13**	26.92	**95.86**	**2.72**	21.15	**78.85**	22.44	**5.77**	86.98	**13.30**	**9.62**
	ROS	532.6	**97.78**	0.80	19.23	**96.30**	**2.24**	21.15	**97.19**	**1.92**	13.46	**97.93**	**0.16**	25.00
80%	IMB	754.9	98.25	0.64	**15.38**	**97.63**	**1.28**	15.38	**97.93**	**0.96**	15.38	97.34	**0.64**	26.92
	RUS	1165.2	**87.87**	**10.90**	26.92	**92.01**	**6.41**	26.92	82.25	18.59	**7.69**	**85.80**	14.42	**11.54**
	ROS	559.6	**96.76**	1.28	26.92	**97.34**	**0.96**	23.08	**97.93**	**1.28**	11.54	**97.63**	**0.32**	26.92

Table 4.7: Classification results while considering the most frequent 100 features that appear in the generated GP models (per class) for 13 Users (HS1)(Experiment. 4).

Rate of Training Samples	***Data-sets***	***SVM***			***k-NN***			***Naïve Bayes***			***Random Forest***		
		VR (%)	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***
60%	IMB	95.72	1.76	34.62	96.60	0.80	34.62	95.71	2.40	26.92	95.41	0.64	51.92
	RUS	82.25	17.47	21.15	93.93	5.13	17.31	77.37	23.56	11.54	86.98	12.98	13.46
	ROS	96.15	0.64	42.31	91.72	7.85	13.46	87.57	12.66	9.62	86.69	13.30	13.46
80%	IMB	97.65	1.28	15.38	97.63	0.96	19.23	97.63	1.60	11.54	97.34	0.64	26.92
	RUS	74.45	25.32	15.38	90.53	8.97	15.38	78.40	22.44	11.54	82.25	18.27	11.54
	ROS	96.75	0.96	30.77	92.31	6.41	23.08	91.42	8.65	7.69	85.50	14.42	15.38

Table 4.8: Classification results based on the average performance of all GP-generated analytical functions (100 analytical functions per class) with various fitness functions using HS1 data set.(Section. 4).

Fit.Func	***60%***				
	VR (%)	***FAR (%)***	***FRR (%)***	***No. of Variables***	***No. of Operations***
NH	91.49	5.65	42.85	7.67	22.63
BH1	89.27	8.44	38.15	6.22	8.46
BH2	89.77	7.92	37.98	6.18	8.26
PA1	89.67	8.00	38.29	6.30	8.57
PA2	89.83	7.94	36.87	6.04	8.23
EXP	89.80	8.01	36.50	6.19	8.65
SNL	89.85	7.84	37.85	5.93	7.75
SL	89.75	7.98	37.56	6.35	8.69

Table 4.9: Classification results based on the average performance of the first quartile of the best GP-generated analytical functions (per class) with various fitness functions for 13 Users (HS1)(Section. 4).

Fit.Func	***60%***				
	VR (%)	***FAR (%)***	***FRR (%)***	***No. of Variables***	***No. of Operations***
NH	95.90	1.93	30.13	7.39	22.88
BH1	92.89	6.34	16.31	5.97	7.74
BH2	91.84	6.56	27.31	5.84	7.32
PA1	91.34	7.19	26.31	6.10	8.14
PA2	93.25	5.90	16.92	5.73	7.87
EXP	93.14	6.17	15.15	6.02	8.05
SNL	93.18	5.98	16.92	5.94	7.86
SL	92.91	6.32	16.35	6.17	8.32

Table 4.10: Classification results while considering the union of all the features that appear in the generated GP models (per class) with various fitness functions for 13 Users. (Section. 4).

Rate of Training Samples	***Fit.-Func.***	***Average No. of Attributes***	***SVM***			***k-NN***			***Naïve Bayes***			***Random Forest***		
			VR (%)	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***
60%	NH	692.9	97.50	0.16	30.77	97.78	0.16	26.92	96.89	0.96	28.85	96.60	0.16	42.31
	BH1	517.77	95.56	0.32	50.85	96.60	1.12	30.77	97.04	0.96	26.92	96.89	0.16	38.46
	BH2	515.5	95.56	0.48	51.92	97.19	0.64	28.85	96.75	1.44	25.00	96.75	0.16	40.38
	PA1	531.2	95.27	0.32	57.69	96.60	1.12	30.77	97.04	1.12	25.00	96.75	0.16	40.38
	PA2	504.0	95.56	0.32	53.85	96.75	1.12	28.85	97.19	0.96	25.00	96.89	0.16	38.46
	EXP	513.46	95.41	0.32	55.77	96.60	1.12	30.77	97.63	0.96	19.23	96.75	0.16	40.38
	SNL	492.08	95.56	0.48	51.92	96.45	1.44	28.85	97.04	1.44	21.15	96.60	0.16	42.31
	SL	535.77	95.56	0.32	53.85	96.45	1.44	28.85	96.60	1.28	28.85	96.30	0.16	46.15

Table 4.11: Classification results while considering the most frequent 100 features that appear in the generated GP models (per class) based on various fitness functions for 13 Users (HS1)(Section. 4).

Rate of Training Samples	***Fit.-Func.***	***SVM***			***k-NN***			***Naïve Bayes***			***Random Forest***		
		VR (%)	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***	***VR (%)***	***FAR (%)***	***FRR (%)***
60%	NH	95.72	1.76	34.62	96.60	0.80	34.62	95.71	2.40	26.92	95.41	0.64	51.92
	BH1	95.41	0.16	57.69	96.30	1.12	34.62	96.30	1.60	28.85	96.15	0.48	44.23
	BH2	95.56	0.16	57.69	97.04	0.64	30.77	96.30	1.76	26.92	96.30	0.32	44.23
	PA1	95.12	0.16	61.54	96.98	1.12	26.92	96.75	1.12	28.85	96.89	0.32	36.54
	PA2	95.56	0.32	53.85	97.04	0.80	28.85	96.75	1.60	23.08	96.60	0.48	38.46
	EXP	96.01	0.00	51.92	97.49	0.32	28.85	96.45	1.60	26.92	96.30	0.32	44.23
	SNL	95.56	0.16	55.77	96.15	0.96	38.46	96.89	1.12	26.92	96.30	0.48	42.31
	SL	95.41	0.00	59.62	97.04	0.80	28.85	96.75	0.96	30.77	96.30	0.64	40.38

Conversely, the number of operations includes all arithmetic operations used in each function (duplication of identical operations are counted)[10].

The average performance over all GP models is considered in Table 4.4 and Table 4.8. The classification performance of the IMB and the ROS data sets is comparable (slightly lower for ROS), but somewhat lower with regards to the classical classifiers. However, for HS1, only a small number of features averaging 7.67 and 6.42 were utilized by GP, when considering the IMB and ROS data sets, respectively. This is in contrast to the original high dimensional feature space of $10,000$ attributes exploited by the reference classifiers as well as the 149.5 (ROS) and 243.1 (IMB) attributes exploited by the reference attribute selection method (subset size forward selection). Similar to previous observations, RUS data set performance degrades, but the amount of available training instances becomes very small (12 and 16 for the 60% and 80% training sets respectively for HS1). The FAR for IMB and ROS sets is better than many of the reference classifier results described in Table 4.2, whereas for the RUS sets, the opposite is observed. Moreover, GP models for IMB and RUS required relatively simple equations (i.e. fewer operations) but that was not the case for RUS. Hence, allowing more complex models, while possibly extending the GEP function set as well, may result in an increase in performance[10].

In Table 4.8, it is observed that there is an additional decrease in the number of variables and in the number of operations with the fitness functions that tackle the problem of unbalanced data sets. GP-generated models using fitness functions (BH1,BH2, PA1, PA2, EXP, SNL, SL) exhibited a decrease in FRR and a slight increase in FAR. However, as FRR of NH is high (42.85%), a decrease would be more favourable. Nonetheless, when the average of the best first quartile models is considered 4.9, an important decrease of FRR can be observed. Moreover, (BH1,BH2, PA1, PA2, EXP, SNL, SL) require a much smaller number of operations 8 compared to those required when the default NH fitness function is used 22.63. A slight decrease in the number of variables is also observed. In addition, the use of fitness functions (BH1, PA2, EXP, SNL, SL) resulted in a substantial decrease of FRR (with the lowest at 15.15%) when our proposed EXP algorithm is used.

The results in Table 4.2 were collected with the optimal choice of parameters for the reference classifiers (something that was not done for GP, as only default parameters were considered). A more balanced comparison can be accomplished by forming an ensemble model using those in the first quartile of the classification accuracy empirical distribu-

tion. Then, the IMB and ROS classification results (Table 4.5) are equivalent to (and in certain cases outperform) those of the reference classifiers and the reference attribute selection method. However, knowing that the results of the reference classifiers are collected using fine-tuned parameters makes GP results more appealing. Furthermore, and very interestingly, GP models consist of only a very small subset of attributes (6.18 to 7.39). This is slightly smaller in comparison to all GP models (Table 4.4). The model selection approach considered utilized the test data as validation, only for illustration purposes. However, future studies will utilize a separate validation data set. The classification performance for the reference classifiers is observed to degrade in certain cases with an increase in the size of the training set (from 60% to 80%) (Table 4.2), whereas for GP (Table 4.4) the classification performance consistently shows improvement, regardless of the sampling type. [10]

In Table 4.10 a decrease in the number of variables used in (BH1,BH2, PA1, PA2, EXP, SNL) is clearly observed. While considering the Naive Bayes classifier, all of the aforementioned fitness function results outperformed those obtained with NH for FRR and VR. When the Random Forest's classifier is considered, all fitness functions exhibited the same FAR but with a slight decrease when (BH1,BH2, PA1, PA2, EXP) are used. Moreover, after applying an extreme feature reduction to only 1% of the original attributes, the generated models based on fitness functions that solve the unbalanced data set problem (BH1,BH2, PA1, PA2, EXP, SNL, SL) outperformed the results obtained with the NH fitness function.

To summarize the above findings, it was observed that GP classification performance is comparable to the performance of the well established reference classifiers. Moreover, solving the unbalanced data set problem with several fitness functions brought better results than those obtained with NH fitness functions. Moreover, GP models utilize on average less than 10 attributes, as opposed to the original 10000 attributes utilized by the reference classifiers.

Feature Reduction

From Table 4.6, it can be observed that when the selection of attributes based on Eq-3.20 ($\mathcal{R}_{V_p,\mathcal{F}_{GP}}$) is utilized in combination with reference classifiers, an increase in the overall classification performance is achieved. In fact, the results obtained in many cases

outperformed the classification results obtained with the original set of $10,000$ attributes (shown in bold characters in Table 4.6). For instance, we can observe a significant improvement in classification results when the k-NN classifier is considered, especially the FAR, which is a very important measure in user identity verification. It is very impressive behaviour since only 5% to 7.5% of the original number of features are utilized, which is a significant dimensionality reduction. Similarly, from Table 4.6, when considering the *ROS_60* data set with the SVM classifier, a 97.78% with a 0.80% FAR is achieved. This result is very promising when compared to previous work [27, 62] where a VR of 98% was achieved but with a FAR of 25%. Moreover, very promising results are obtained when considering a small subset of features that represents only the most frequent features extracted from generated GP models (Table 4.7) and when choosing the first quartile of the best GP models (Table 4.5). Comparable results were achieved with a similar dimensionality reduction when solving the unbalanced data set problem through the introduction of several fitness functions (Table 4.10,Table 4.11)[10].

Furthermore, in a severe feature reduction experiment (Table 4.7 and (Table 4.11 where only 1% of the original attributes were selected, the classification performance only decrease slightly. In this case, the dimensionality reduction rate can be variable, giving room to choose the desirable reduction rate. However, the selected rate still has a higher number of attributes selected in comparison to the models obtained by the GP (Section 4.3.3 and Tables 4.4, 4.5)[10].

Relevance of Haptic Data Types in Identity Verification

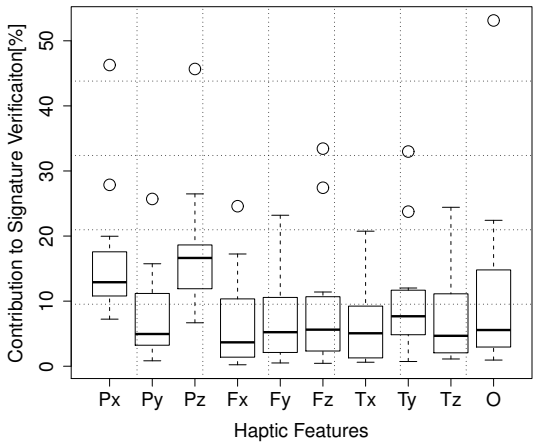
Figs. 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, 4.13, 4.14, 4.15 present box plots that represent the frequency of haptic data types that are considered by GP-generated models when IMB,RUS and ROS data sets, BH1,BH2, PA1, PA2, EXP, SNL and SL fitness functions. Each box has lines at the lower quartile, the median, and the upper quartile values. Whiskers, which are the lines extending from the end of each box, are shown to present the extent of the frequency of different haptic data types found in the generated GP-models. Circle symbols 'o' that are beyond the box and whiskers lines represent outliers. It can be noticed that the frequency plot of RUS data sets differ significantly from those associated with IMB,ROS, BH1,BH2, PA1, PA2, EXP, and SNL. More importantly, RUS classification results in this experiment were significantly low as illustrated in Tables 4.4 and 4.5, thus the RUS frequency plot is unreliable. Moreover,

in Figs. 4.6,4.8, 4.9, 4.10, 4.11, 4.12, 4.13, 4.14, 4.15, it can be observed that position information (P_x, P_y, and P_z) utilized by GP-models with a probability of approximately 0.5. Thus, all remaining haptic data types (i.e., force, torque and orientation (F_x, F_y, F_z, T_x, T_y, T_z, and O) were utilized by GP-models with a probability of approximately 0.5. This is a promising result as it shows the importance of haptic data types, as they were used approximately as frequently as the visual features where visual features are utilized solely by traditional handwritten signatures[10].

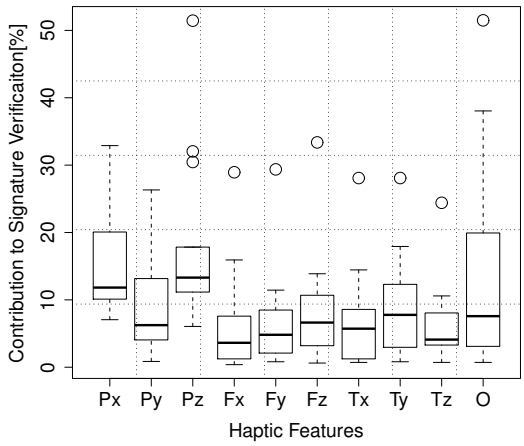
It can be observed that the interquartile range in Figs. 4.9, 4.10, 4.11, 4.12, 4.13, 4.15, are more condensed in comparison to Fig. 4.6, suggesting an agreement among generated models on haptic feature significant when fitness functions are designed to solve the unbalanced data set problem.

4.4 Conclusion

The first stage represents a pre-determined task where a user is required to virtually sign a check. The user is required to mimic the original signature to be authenticated. Several attribute selection methods are utilized including the proposed two feature selection methods based on GP. In addition, several fitness functions have been proposed to solve the unbalanced data set problem. The generated analytical functions maintained comparable classification success rate in addition to huge dimensionality reduction.

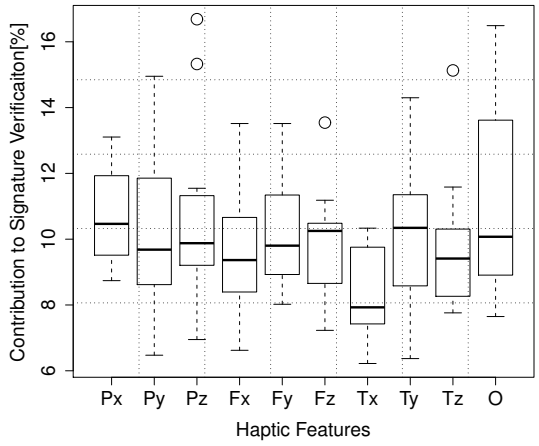


(a)

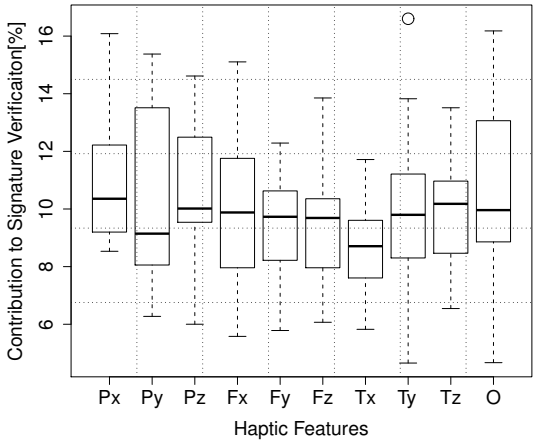


(b)

Figure 4.6: Box plots illustrating the frequency of haptic data types found in GP-generated models when unbalanced datasets are considered with (a) 60% training data, (b) 80% training data.(Section. 4)

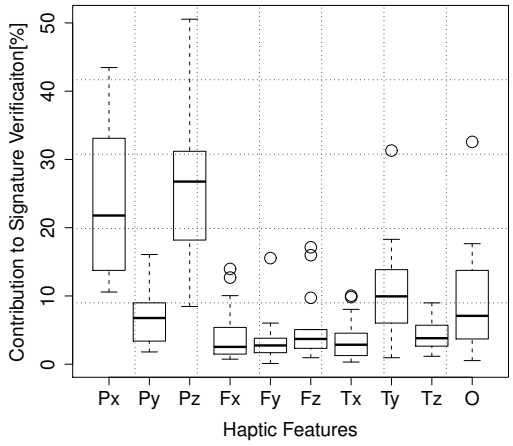


(a)

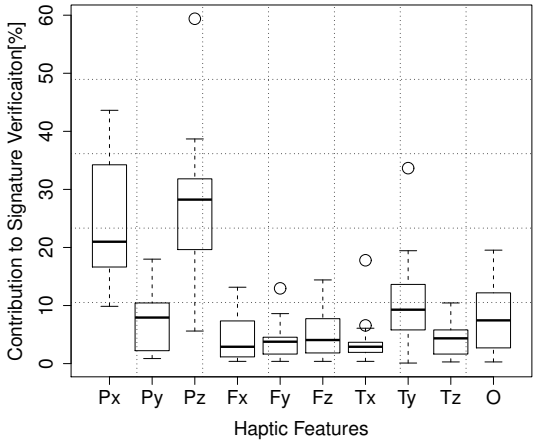


(b)

Figure 4.7: Box plots illustrating the frequency of haptic data types found in GP-generated models when RUS datasets are considered with (a) 60% training data, (b) 80% training data.(Section. 4)

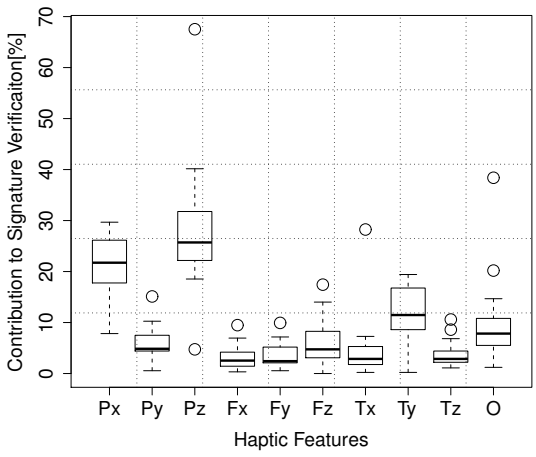


(a)



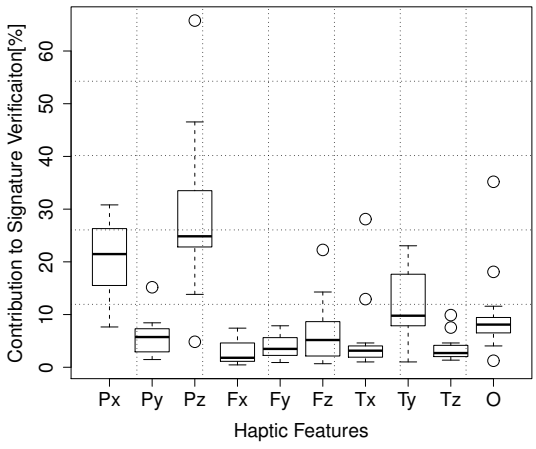
(b)

Figure 4.8: Box plots illustrating the frequency of haptic data types found in GP-generated models when ROS datasets are considered with (a) 60% training data, (b) 80% training data.(Section. 4)



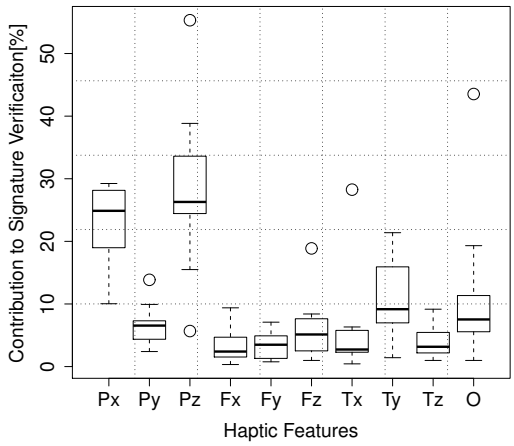
(a)

Figure 4.9: Box plots illustrating the frequency of haptic data types found in GP-generated models when equation 2.2 are considered as a fitness function with (a) 60% training data(Section. 4)



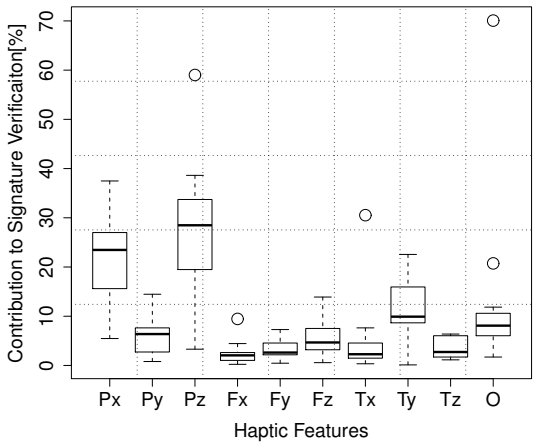
(a)

Figure 4.10: Box plots illustrating the frequency of haptic data types found in GP-generated models when equation 2.3 are considered as a fitness function with (a) 60% training data(Section. 4)



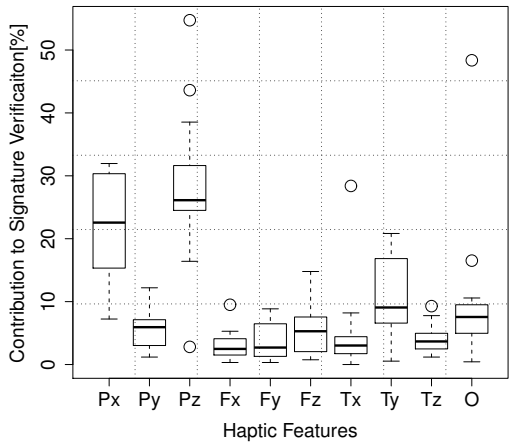
(a)

Figure 4.11: Box plots illustrating the frequency of haptic data types found in GP-generated models when equation 2.7 are considered as a fitness function with (a) 60% training data(Section. 4)



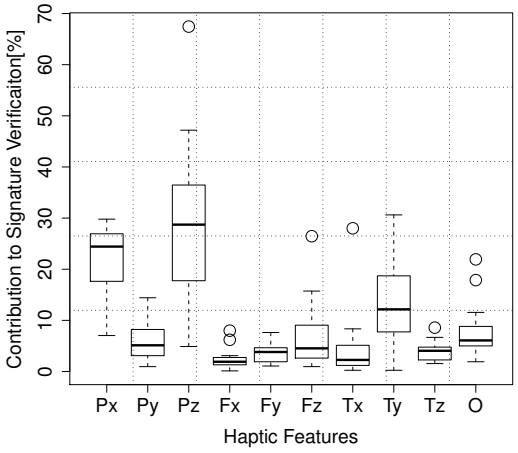
(a)

Figure 4.12: Box plots illustrating the frequency of haptic data types found in GP-generated models when equation 2.8 are considered as a fitness function with (a) 60% training data(Section. 4)



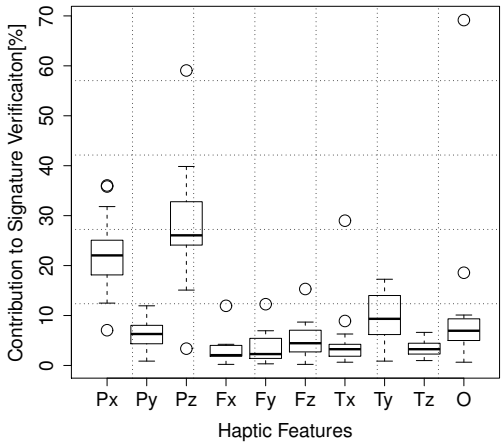
(a)

Figure 4.13: Box plots illustrating the frequency of haptic data types found in GP-generated models when the proposed equation 3.24 are considered as a fitness function with (a) 60% training data(Section. 4)



(a)

Figure 4.14: Box plots illustrating the frequency of haptic data types found in GP-generated models when the proposed equation 3.25 are considered as a fitness function with (a) 60% training data(Section. 4)



(a)

Figure 4.15: Box plots illustrating the frequency of haptic data types found in GP-generated models when the proposed equation 3.26 are considered as a fitness function with (a) 60% training data(Section. 4)

Chapter 5

Semi-Random Tasks in PROFECE: Drawing Experiment

In this chapter, we will present the second stage of our planned experiments (1.1) which consider semi-random movements. As we advance from one experiment to another, the complexity of the problem escalates in terms of freedom of movements.

In this experiment, the user still has to draw a landscape picture. However, the order and the shapes of the strokes are not required to authenticate the user. This experiment tries to capture user behaviour while drawing. In this chapter, we present our experimental setup and procedure. The data set collected from the drawing experiment is utilized for user identification and identity verification. We then present our results.

5.1 Design and Architecture

The identity verification process for user drawings with haptic information relies on the time varying approach (Section. 3.5.2). User drawings can be seen as semi-random and semi-continuous tasks, where the user is required to draw a landscape picture that spans over a long period of time. Therefore, the aforementioned approach suits this task since it allows the dependency of a selected time period to identify the user in question, rather than waiting for the whole drawing to be completed, which is required by the haptic feature dimensionality reduction approach (Section. 3.5.2). The high frequency and richness of haptic information resulted in a large data set. In this section, we describe the

Figure 5.1: A Snapshot of the drawing environment where one monitor's screen shows the required picture to draw and the other screen shows the user's drawing through the haptic device[8]

design and architecture of the proposed solution for user authentication drawing with haptic information.

The overall architecture of haptic-enabled drawing application is similar to Fig. 4.2. The user uses the haptic-enabled virtual drawing application to draw a pre-defined landscape, and the observer component observes every detail of the users sketch, including haptic features such as three-dimensional position, force, etc., and passes it to the verifier. The verifier component analyzes the captured sketch and tries to match it to the legitimate user's sketches, to either grant or decline access to the protected system.

Figure 5.2 shows a simple state chart diagram that summarizes, in three steps, the user authentication process of user drawings with haptic information. It starts with requesting the username of the user. The user is then asked to draw the presented picture, and the system either grants access or requests that the user repeat the process because of a mistake in the provided username or the recorded user drawing. The "verifying identity" state is expanded in Fig. 5.3. This expansion shows several steps and several branches to solve the user authentication problem based on user drawings with haptic information. After the user sketches the pre-defined landscape, the verification process performs a feature generation that is described further in Section. 5.1.1. In the case where it is a verification problem, the data set is converted from a multi-class problem

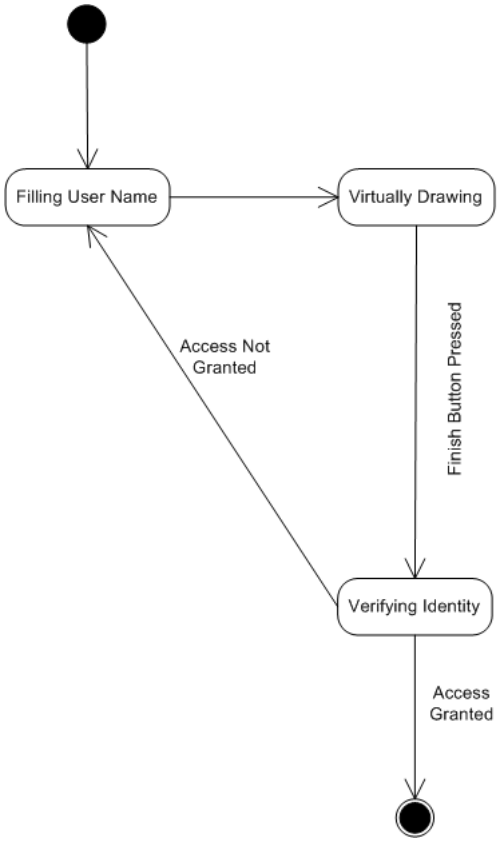


Figure 5.2: Simple state chart diagram of user authentication using haptic-enabled virtual drawing application.

to a two-class problem. Afterwards, as we follow a hold-out test data evaluation in order to have a testing set that is independent from the training data set, the data sets are divided into training and testing data sets. The Neural Network classifier is then utilized for classification in addition to applying the weighting factor described in (Section. 3.5.2). Finally, the system performs the evaluation process and decides if the user is granted access or denied.

The class diagram of the proposed solution for solving the user authentication problem based on user drawings is presented in Fig. 5.4. The observer component observes the users sketch, including the visual and haptic information. The verifier component bases its decision on several other components depending on the configuration requested, such as whether to consider an unbalanced data set solution or to use a certain classifier. The FeatureGenerator filters the original features and generates features as described

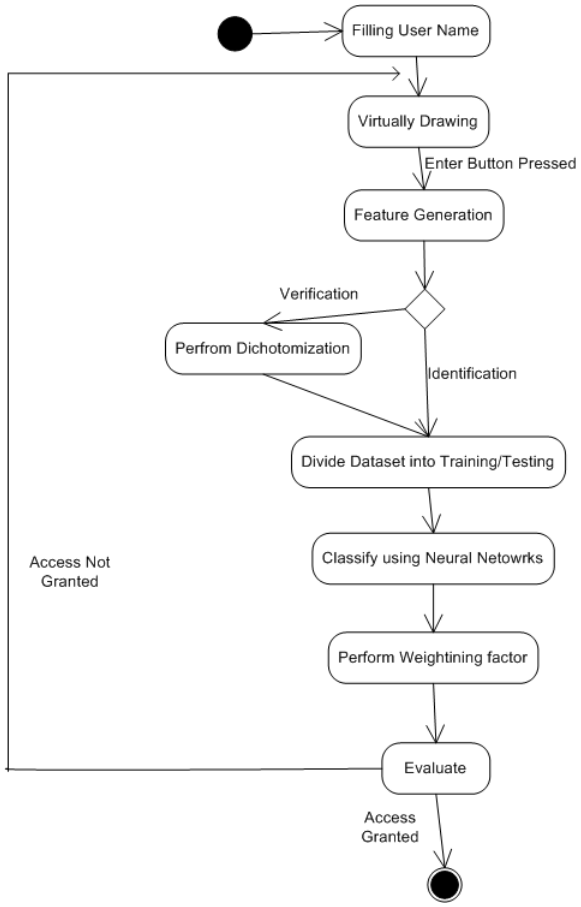


Figure 5.3: Extended state chart diagram of user verification using haptic-enabled virtual drawing application.

in Section. 5.1.1. The WeightingFactorManager component is responsible of assigning a lower weight for the majority class and a higher weight to the minority class. The Evaluator evaluates the classifier on concerns and reports the classification results.

5.1.1 Feature Generation

When a user virtually draws, we capture the three dimensional velocity v, force f, and angular rotation a of a haptic device tooltip. Instead of using the above raw features directly as described in [7], and following the Time Varying approach (Section. 3.5.2), a filtering process is performed that generates the mean values (μ), and the variance values (σ^2) of each feature, at every 100 millisecond. The filtering process reduces the noise in the data sets and is considered a lossy compression. Thus, an instance in the drawing

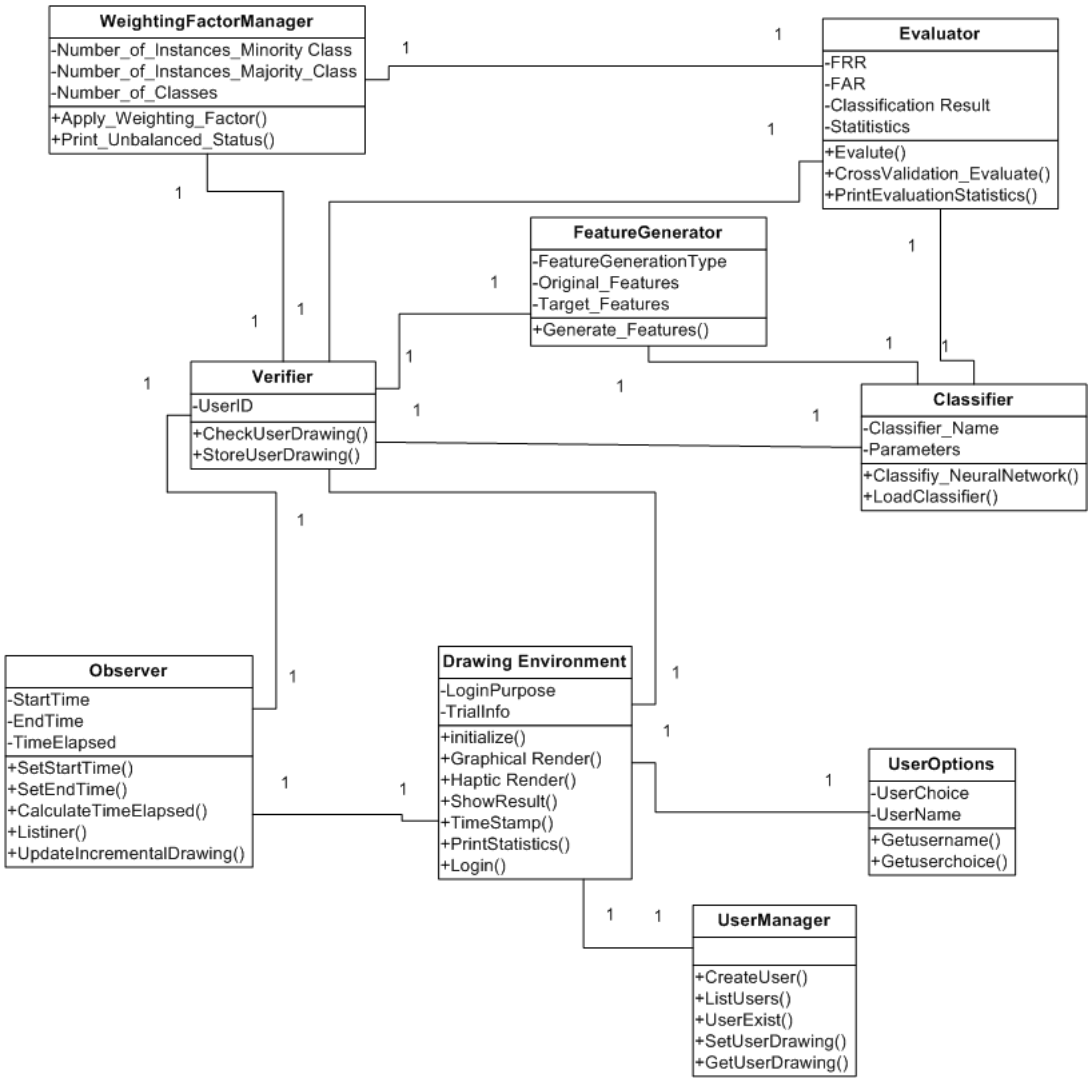


Figure 5.4: Class diagram of Identity verification using haptic-enabled virtual drawing application.

system is represented as:

$$\begin{aligned} I &= \mu(v_x), \sigma^2(v_x), \mu(v_y), \sigma^2(v_y), \mu(v_z), \sigma^2(v_z), \mu(f_x), \mu(f_y), \mu(f_z), \sigma^2(f_z), \\ &\quad \mu(a_x), \sigma^2(a_x), \mu(a_y), \sigma^2(a_y), \mu(a_z), \sigma^2(a_z) \end{aligned}$$

5.1.2 The Identification and Verification of Users based on their Haptic-enabled Drawing

We address the problem of identifying and verifying users based on their haptic information while they sketch a pre-defined landscape. In the identification process, we use a multilayer perceptron neural network (MLP NN) [74] with set I as an input. The set I consists of 16 input neurons connected to hidden layers of 15 neurons. In addition, the output layer consists of 14 neurons that correspond to each user. The neurons are connected to the hidden layers. In the neural network, we use a supervised learning approach with back propagation algorithms with a momentum of 0.2 and learning rate of 0.3. All analysis is performed with the support of the Weka machine learning tool [74]. In the verification process, we use MLP NN with set I as an input. The set I consists of 16 input neurons connected to hidden layers of 9 neurons. Moreover, the output layer consists of two neurons that are connected to the hidden layers. Here, we use a supervised learning approach with back propagation algorithms, a momentum of 0.2, and learning rate of 0.3.

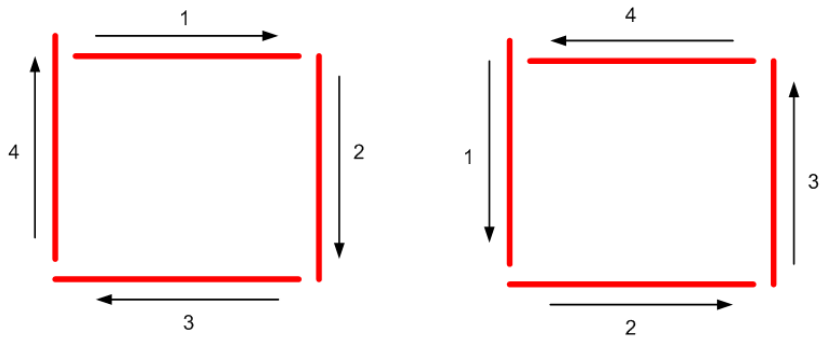
5.2 Method

5.2.1 Participants

A total of fourteen users volunteered to participate in this drawing experiment. A total of twelve participants are males and two are females, of ages ranging from 25 to 35. The participants were briefed about the experiment procedure and had the choice to withdraw whenever they felt it was necessary. All subjects proceeded without any hesitation.

5.2.2 Apparatus

For the drawing experiment, a picture of a vehicle in front of a natural scenery background is presented to the subjects. Each user is required to naturally draw the picture in the virtual space with the use of a haptic device. There was no information given about what part of the scene to draw, or to what extent the details should be present. The users draw on a virtual plate in a 3-D virtual environment, developed in DISCOVER Lab, where the tooltip of the Phantom OMNI device [68] can be grabbed and moved as shown in Fig. 5.1. The moment the tooltip of the haptic device comes in contact with



(a) Clockwise direction (b) Counterclockwise direction

Figure 5.5: An illustration of two different ways of drawing a rectangle. The numbers 1,2,3 and 4 depict the order of strokes and the arrows illustrate the stroke's direction.[8]

the virtual plate, the user senses a force feedback against the plate, originating from the Phantom OMNI device, and a visual impact represented by red dots drawn in the contact area on the screen[8].

5.2.3 Procedure

The drawing experiment is a step forward in order to continuously authenticate a user while a given task in a haptic application is performed. In order to authenticate a user, a user has to draw a landscape picture (can be replaced with any picture). To be authenticated, this task can be considered as a semi-random task in terms of the user's freedom of movements. It is considered semi-random because even if we ask a user to draw a simple shape such as a rectangle several times, the subject might draw it in several different ways. A simple rectangle can be constructed with variations in stroke direction and order, as depicted in Fig. 5.5(a) and Fig. 5.5(b). Consequently, drawing a more complex landscape which consists of several objects can be performed in a high number of different ways as observed in Fig. 5.6(a) and 5.6(b) where a user was trying to draw the same picture twice. Fig. 5.6 shows some details drawn by the user that are not present in the second drawing, such as the door knobs, car details, buildings, etc. In order to show the differences in more detail, a zoom-in of Fig. 5.6(a) and (b) is illustrated in Fig. 5.6(c),(d),(e), and (f). This observation lead us to the importance of capturing

the user's behaviour instead of focusing only on the direction and order of the strokes [8].

At the beginning of the experiment, we gave an explanation about the system and the experiment's intended goal and procedure. Then, users were free to try the system by freely sketching any shape. The users were free to draw until they felt comfortable using the haptic device. Some subjects were familiar and had been previously exposed to a haptic environment; in a short period of time, they were ready to begin the experiment. As soon as the subjects were ready to start the experiment, we showed them the picture of a car with a landscape background as shown in Fig. 5.1, and they were requested to use the haptic device to sketch the picture twelve times. The subject reports when a draw is completed, and then the haptic data is stored and considered a trial[8]. Moreover, subjects had a break to relax in order to avoid haptic fatigue as described in [39].

In the drawing experiment, the haptic data set is divided into two sets. A data set which contains the first six trials (t_1 to t_6) where it is utilized for training purposes and a dataset which contains trials (t_7 to t_{12}) (hold-out data set) which are reserved for testing purposes[8].

5.3 Drawing Experimental Results

The drawing experiments were conducted as a middle stage, to study the way toward fully random free movements, represented in the haptic-enabled interpersonal communication system (Section 6). In this experiment, we follow Approach 2 (Section 3.5.2), including the consideration of the unbalanced data set problem. Since the instances that resulted from user drawings vary with each user and each trial.

5.3.1 Identification

In this section, we start presenting the results of user identification. Fig. 5.7 presents the overall average of the users identification success rate for all trials t_7.. t_{12} on all users u_1.. u_{14} which resulted in a classification success of 55.9% for (Original) and 75% for (O+WF) where (O+WF). "Original" represents the classification results of the original data set without the introduction of any solution to the unbalanced data set problem, while "O +

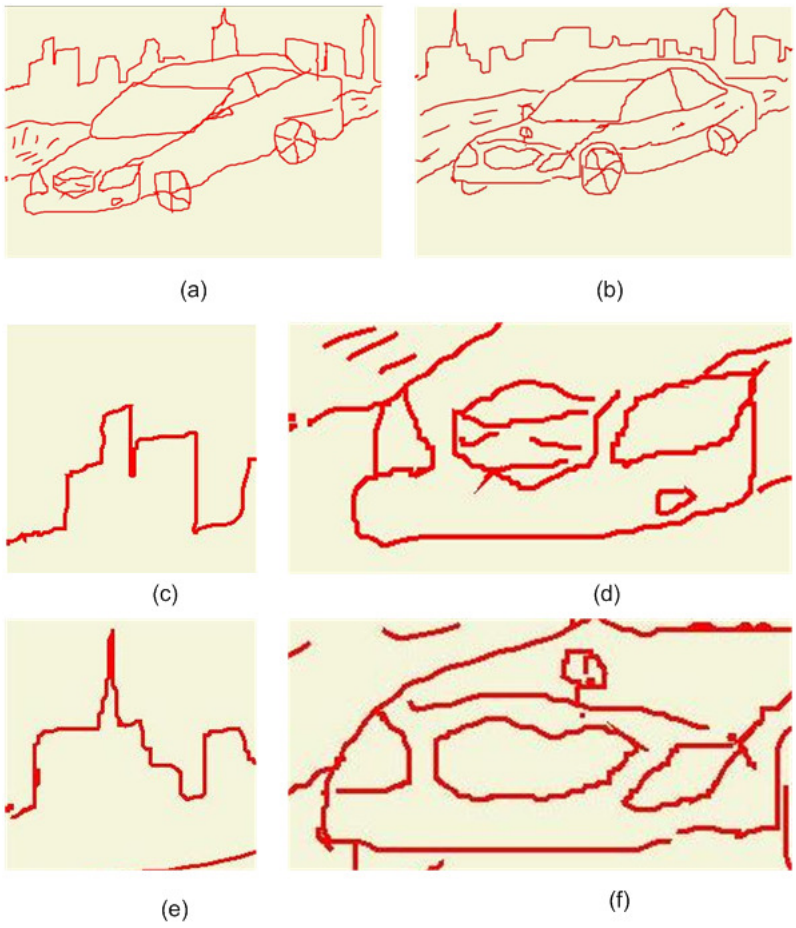


Figure 5.6: Two drawings of the same scene for the 10th user in (a) and (b). The difference on the details of drawings is clearly visible, in addition to the differences in strokes order. (c) and (d) are a zoom-in parts from (a) while (e)(f) are a zoom-in parts of (b)[8]

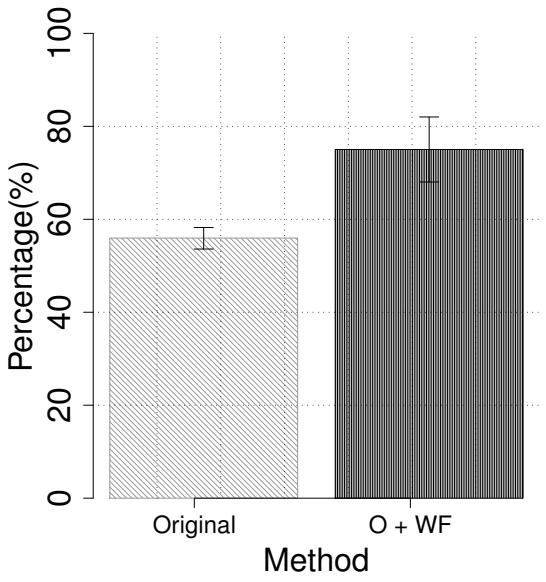


Figure 5.7: Overall Average User Identification Rate for all users u_1.. u_{14} on trials t_7.. t_{12} based on the discussed two methods. whiskers illustrate the standard error.(Section. 5)

Table 5.1: User verification results shows the FAR and FRR when a user-dependent threshold is utilized and adjusted based on trials t_7.. t_9.(Section. 5)

Method	Original		O+WF	
Time	FAR	FRR	FAR	FRR
10s	28.29%	31.29%	31.21%	31.43%
20s	23.43%	27.29%	29.14%	23.86%
30s	30.50%	17.93%	26.57%	20.29%
40s	37.36%	11.86%	24.86%	13.86%
50s	34.71%	11.50%	25.64%	10.29%
60s	33.07%	14.71%	25.50%	12.86%
MaxTime	33.21%	2.43%	22.86%	3.57%

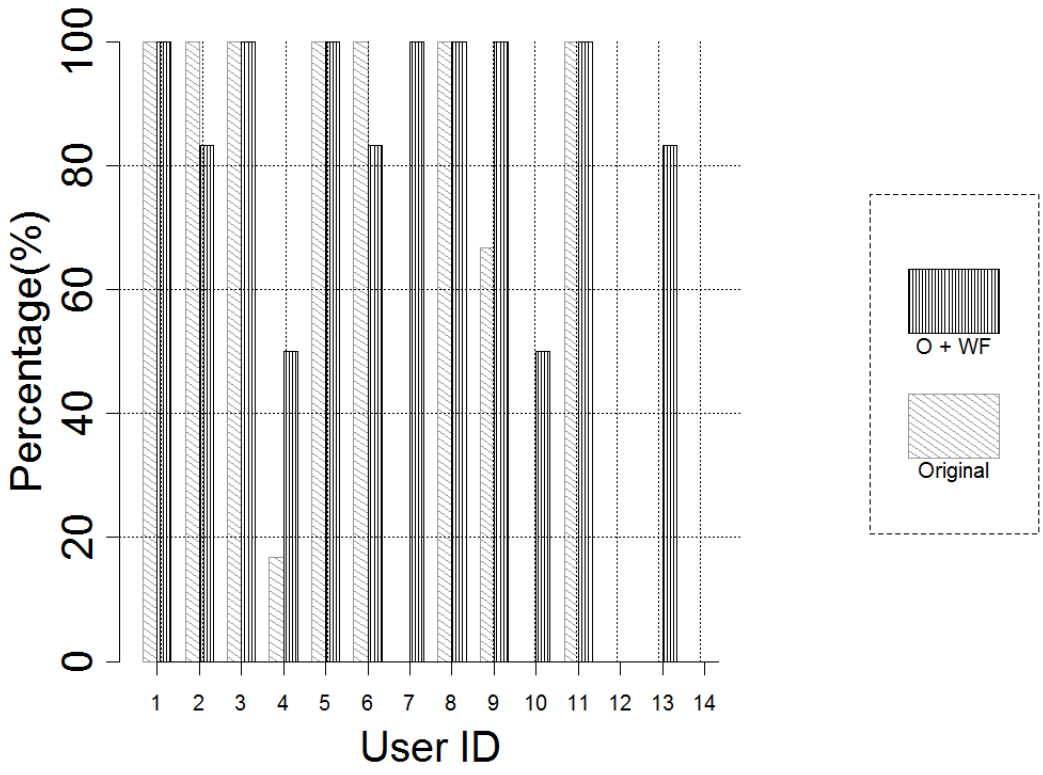


Figure 5.8: Average User Identification Rate for all trials t_7.. t_{12} on users u_1.. u_{14} based on the discussed two methods(Section. 5)

Table 5.2: User verification results shows the FAR and FRR when a user-dependent threshold is utilized and adjusted based on trials t_7.. t_{12}.(Section. 5)

Method	Original		O+WF	
Time	FAR	FRR	FAR	FRR
10s	19.00%	37.79%	12.93%	45.71%
20s	18.79%	29.07%	20.14%	25.64%
30s	26.29%	18.71%	20.29%	21.07%
40s	25.21%	14.57%	18.57%	14.14%
50s	25.00%	13.86%	20.71%	9.71%
60s	23.71%	14.71%	19.29%	12.29%
MaxTime	22.57%	4.79%	15.79%	4.79%

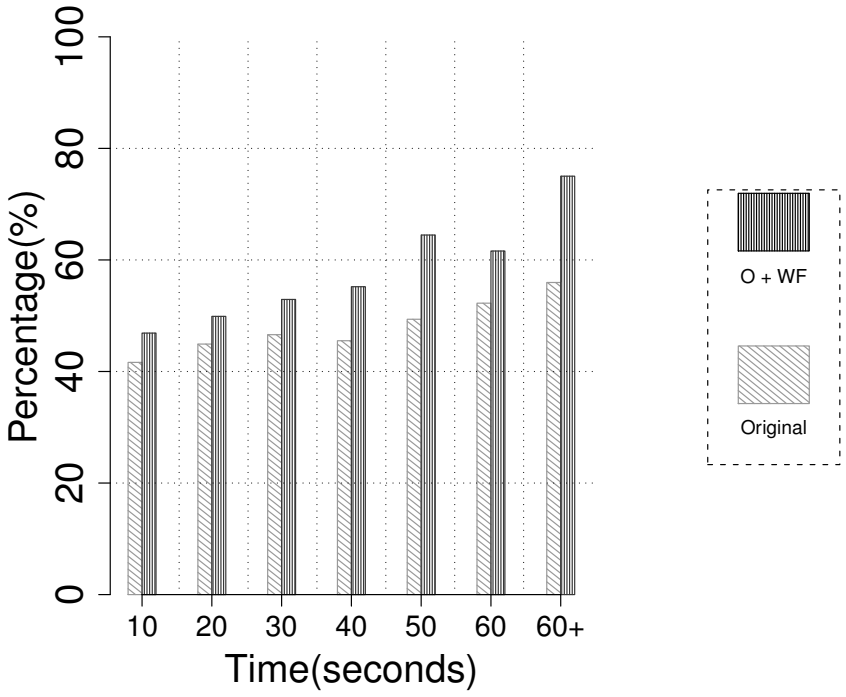


Figure 5.9: Overall Average User Identification Rate for all users u_1.. u_{14} on all testing trials t_7.. t_{12} while performing the identification process in every 10, 20, 30 ,40, 50, 60 seconds, and a complete trial (60+) based on the discussed two methods(Section. 5)

WF" represents the classification results after the introduction of the weighting factor as described previously (Section 3.5.2). As can be observed, there is a significant improvement, from 55.9% (Original) to 75% (O + WF). This improvement is clearly observed by looking into the classification success rate of each user (Fig. 5.8). Subjects u_1,u_2, u_3, u_5, u_6, u_8, and u_{11} achieved 100% success rates while subjects u_{12}, u_{13} and u_{14} have 0% success rates without considering the weighting factor. However, after a closer look into users (u_4, u_7, u_9, u_{10}, u_{12}, and u_{13}) data set, we observe that their drawings suffered from the problem of a relatively small number of instances collected in comparison to the other users, because they drew the landscape quicker than the rest. In addition, the degree of unbalance fluctuates among users. User u_7, u_{12}, and u_{13} have much smaller data sets of around 40 seconds, compared to others (80 seconds, and more). The utilization of the weighting factors improved the classification success rate for u_7 and u_{13}, from 0% and 0% to 100% and 83.3%, respectively. However, this improvement sacrificed the classification results of some of the other users. Moreover, the classification results of u_{12} and u_{14} did not improve, and we need to investigate what other factors might affect this poor result in a future work [10].

In order to examine the system's performance considering smaller time segments rather than analyzing a complete trial, we split the testing trials into time frames of 10, 20, 30, 40, 50, 60, and 60+ seconds, where 60+ represents the entire trial for identification, with no time frame segmentation. As discussed earlier, because of the unbalanced data set problem, some users sketched the picture in only 40 seconds; therefore, a further split of 50 and 60 seconds does not apply to those users. However, we utilized the aforementioned users entire trial even if we increased the time segment to 50, and 60 seconds. When a complete trial is considered an average, an identification success rate of 75% is achieved. However, as the time frame shortened, the identification results decreased accordingly, as shown in Fig. 5.9[10].

5.3.2 Verification

The results of the identity verification are presented based on the average False Acceptance Rate (FAR) and the average of False Rejection Rate (FRR). In the identity verification process, we show the results based on the "Original" data set captured during the experiment, and we consider solving the unbalanced data set problem "O + WF"

following Approach 2 (Section 3.5.2).

In addition, a user-dependent threshold is applied where the first six trials are utilized for training, and trials $t_7..t_9$ are used for adjusting the proper threshold. Table 5.1 presents the FAR and the FRR for the verification process, based on the "Original" data set and "O + WF". A False Acceptance Rate of 22.86% @ 3.57% FRR where the system verifies a user only at the triall completion, where O+WF is considered as well. However, as our main target is reaching a continuous authentication system, the identity verification process is applied every ten seconds, from 10 to 60 (Table 5.2). As the time frame expands, a more desirable decrease of FAR and FRR occurs, where it starts as high as 28.29% @ 31.29% FRR, down to 33.21% FAR @ 2.43% when the "Original" data set is considered (the behaviour is similar for O+WF results). The outcome of solving the unbalanced data set problem is clearly seen by the decrease of FRR results when the O+WF method is utilized. Such outcomes are caused by the O+WF method, which assigns the minority class with a higher weight and assigns the majority class with a lower weight [10].

The verification results shown in Table 5.2 are similar to [60], where a user-dependent threshold is utilized after using the first six trials for training and the remaining trials for setting the user-dependent threshold. Even though the result is biased due to threshold adjustment based on the same test data, it is reported just for comparison. As reported previously in Table 5.1, Table 5.2 shows a similar behaviour of a decrease in FRR results when the O+WF method is used, and a decrease in both FAR and FRR when considering a longer time frame [10].

5.3.3 Conclusion

User authentication based on haptic information in drawings is considered a transitional stage towards fully random movement represented in haptic-enabled interpersonal communication system. The second stage as presented in this chapter is not feasible in real life day-to-day user authentication, as the time required to perform the task is considerably long. Therefore, only Neural Networks has been utilized as a classifier. Moreover, some lessons have been learned in this stage, such as imposing a limitation on the time required to achieve a task on the training samples collected, to avoid the unbalanced data set problem.

Chapter 6

Support of Random Tasks in PROFECE: Haptic-Enabled Interpersonal Communication System

In this chapter, we will present the third and final stage of our planned PROFECE experiments (Fig. 1.1). As we advance from one experiment to another, the complexity of the problem escalates in terms of freedom of movements.

The haptic-enabled interpersonal communication system experiment is based on fully free user movements. It is a reflection of a real life situation, where a user is communicating with a remote person via a haptic-enabled interpersonal communication system. Therefore, the user authentication process is continuous as long as the communication channel is open. In this chapter we describe the design and architecture and the experimental settings. We then discuss the experimental results.

6.1 Design and Architecture

The identity verification process for haptic-enabled interpersonal communication systems relies on the time varying approach (Section. 3.5.2). As users interact and communicate with the remote party, it can be considered as a totally random and continuous task where the user communicates and interacts naturally, with no restrictions imposed. It

Figure 6.1: A Snapshot of the haptic-enabled Interpersonal Communication System where the user is asked to use the haptic device in addition to mouse and keyboard to communicate with the remote party

might also span over a long period of time. Therefore, the aforementioned approach suits this task since it allows the dependency on a variable time period, chosen by the user, to identify the user in question, rather than waiting until the end of the session, as required by the haptic feature dimensionality reduction approach (Section. 3.5.2). Moreover, the high frequency and richness of haptic information gathered during the communication session resulted in a large data set. In this section, we describe the design and architecture of the proposed solution for user authentication on haptic-enabled interpersonal communication systems.

The overall architecture of haptic-enabled interpersonal communication systems is similar to Fig. 4.2. The user uses the haptic-enabled interpersonal communication sys-

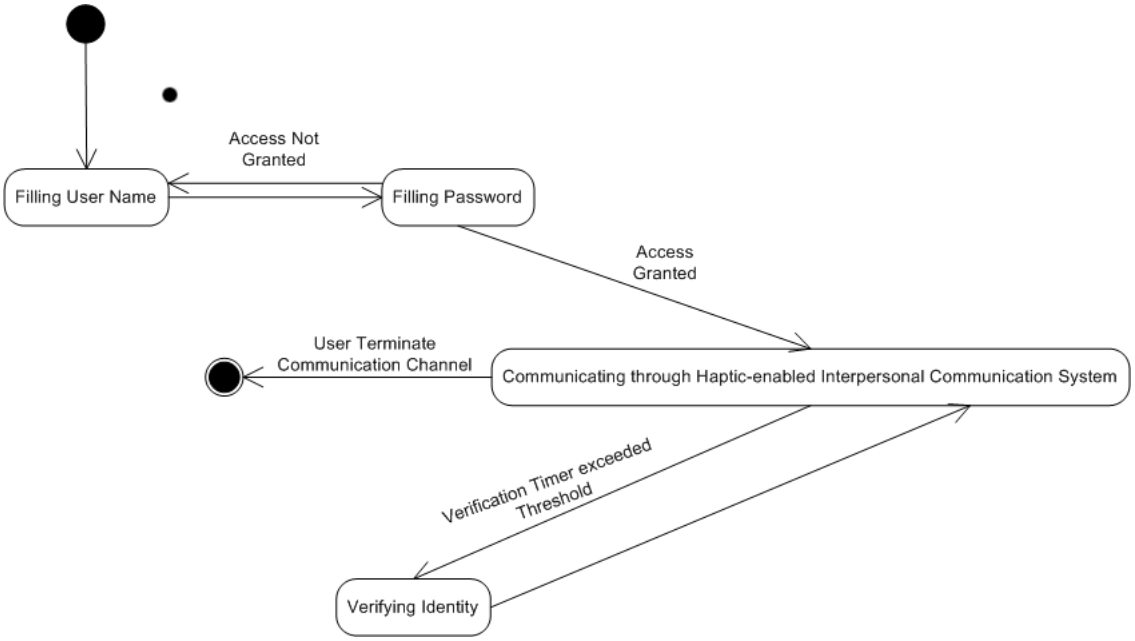


Figure 6.2: Simple state chart diagram of user authentication using haptic-enabled interpersonal communication system.

tem to communicate, interact, and touch the remote party with no restrictions implied, while the observer component observes every detail of the user movements, including haptic features such as three-dimensional position, force, etc., and passes it to the verifier. The verifier component analyzes the recorded user movements and tries to match it to the legitimate user's pattern of movements, to either raise a warning of intrusion, or signal safety ensured to the remote user.

Figure 6.2 shows a simple state chart diagram that summarizes the user authentication process while a user communicates and interacts with the remote party. First of all, it starts by requesting the username. The system then either grants access or requests that the user repeat the process because of a mistake in the provided username or password. If the user is granted access to the haptic-enabled interpersonal communication system, a session will be established with the remote party. Based on the selected verification timer threshold selected, the "verifying identity" state is provoked to periodically verify the user identity.

The "verifying identity" state is expanded in Fig. 6.3. This expansion shows sev-

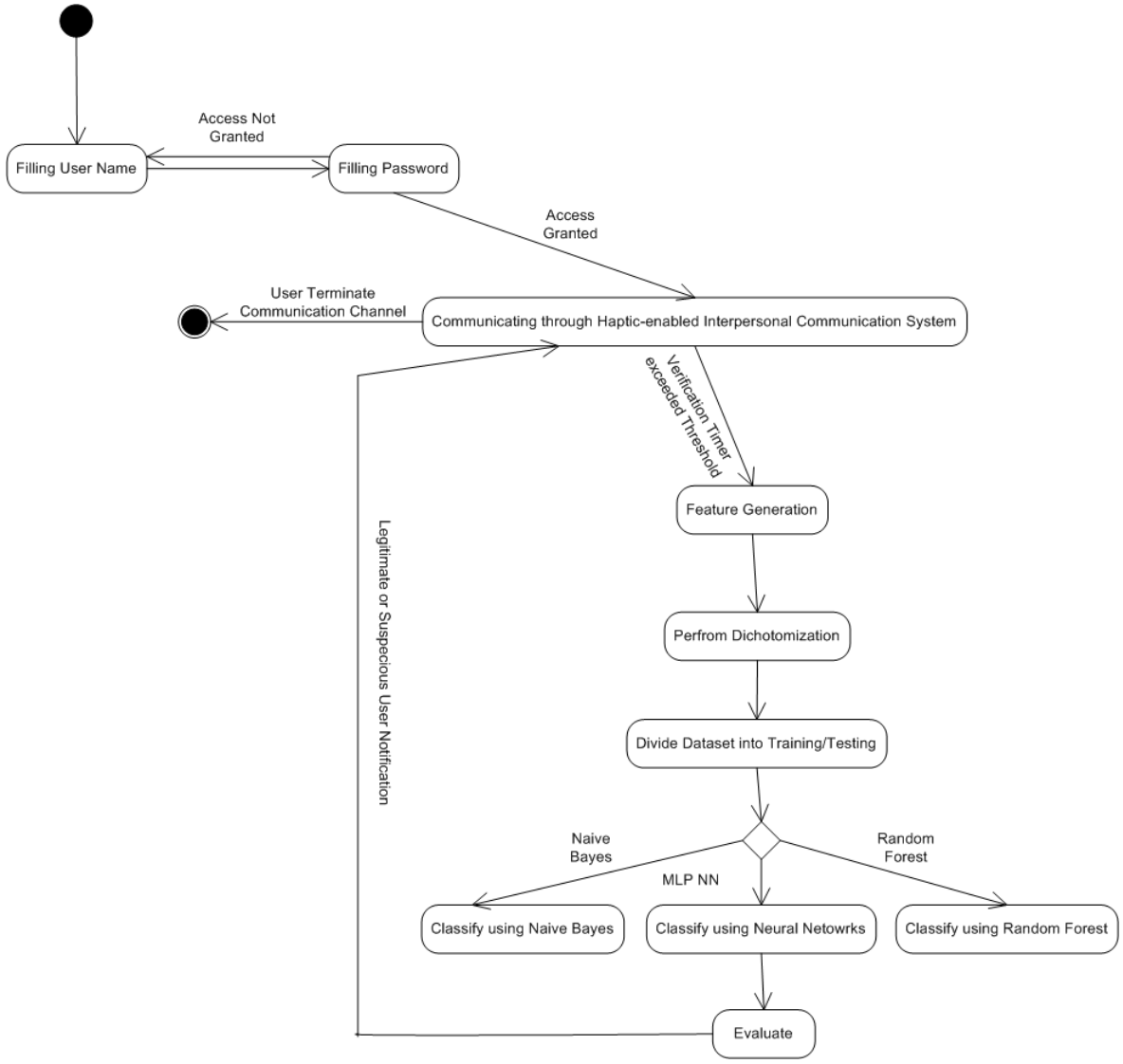


Figure 6.3: Extended state chart diagram of user verification using haptic-enabled interpersonal communication system.

eral steps and several branches to solve the user authentication problem based on user movements in a haptic-enabled interpersonal communication system. While the user interacts with the remote party and after a periodic request to continuously verify the user identity, the verification process performs a feature generation that is further described in Section. 6.1.1. Afterwards, the data set is converted from a multi-class problem to a two-class problem. Then, as we follow a hold-out test data evaluation in order to have a testing set that is independent from the training data set, the data sets are divided into

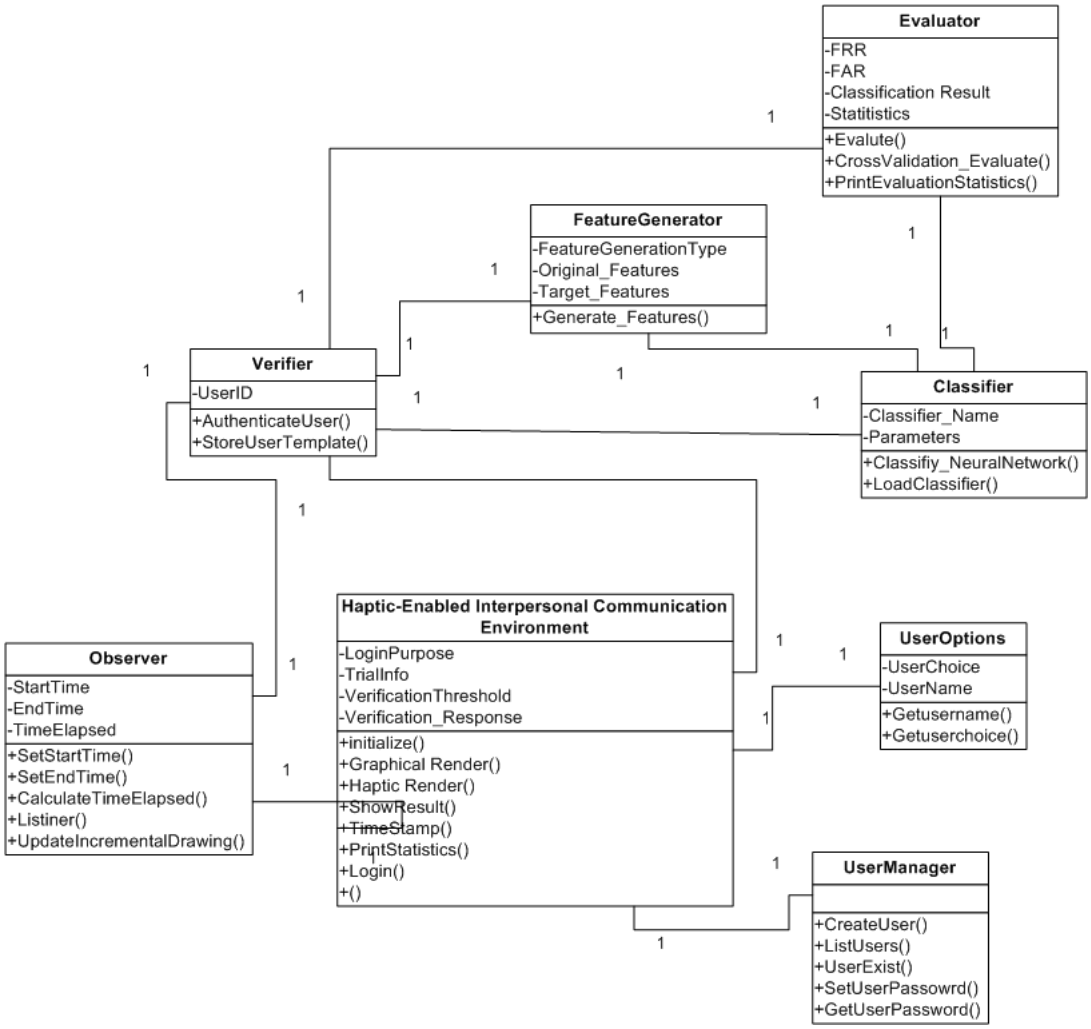


Figure 6.4: Class diagram of identity verification using haptic-enabled interpersonal communication system.

training and testing data sets. The specified classifier then performs the classification process. In this experiment, we ensured an equal number of instances per trial among all users in the training process. Afterwards, the system performs the evaluation process and decides if the user movements match the legitimate user or not. A notification of the verdict is sent to the user. The final decision of whether to continue the session or suspend it is made by the user.

The class diagram of the proposed solution to solve the user authentication problem based on user movements and interactions is presented in Fig. 6.4. The observer component observes the user movements based on the recorded visual and haptic information.

The verifier component bases its decision on several other components, based on the configuration requested of whether to consider a Neural Network classifier, Naive Bayes or Random Forest. The FeatureGenerator filters the original features and generates features as described in Section. 6.1.1 . The evaluator evaluates the concerned classifier and reports the classification results.

6.1.1 Feature Generation

When a user touches the virtual 3-D live object, the system captures the three dimensional position p, force f, velocity v, angular rotation ar, and axis a of a haptic device tooltip. the system follows Approach 2 (Section. 3.5.2), where rather than using the above aforementioned raw features directly, as described in [7], a filtering process is utilized in which it generates the mean values (μ), and the variance values (σ^2) of each feature, every 10 millisecond, as proposed in [9]. The filtering process which is a lossy compression and is utilized to reduce the noise in the data sets. Therefore, an instance in the haptic-enabled interpersonal communication system can be represented as:

$$\begin{aligned} I \;=\; & \mu(p_x), \sigma^2(p_x), \mu(p_y), \sigma^2(p_y), \mu(p_z), \sigma^2(p_z), \mu(v_x), \sigma^2(v_x), \mu(v_y), \sigma^2(v_y), \\ & \mu(v_z), \sigma^2(v_z), \mu(ar), \sigma^2(ar), \mu(a_x), \sigma^2(a_x), \mu(a_y), \sigma^2(a_y), \mu(a_z), \sigma^2(a_z), \\ & \mu(f_x), \sigma^2(f_x), \mu(f_y), \sigma^2(f_y), \mu(f_z), \sigma^2(f_z) \end{aligned} \tag{6.1}$$

6.1.2 The Identification and Verification of Users based on their Haptic Interaction

During user haptic interaction on a haptic enabled interpersonal communication system, our system captures the users movements and their corresponding haptic features in order to identify and verify the users identity. In order to achieve this task, we utilize three different classifiers. First, we use a multilayer perceptron neural network (MLP NN) (Section. 3.3.4) with the following specifications. A set I of 26 input neurons connected to hidden layers of 20 neurons. Moreover, the MLP NN consists of output layers of fourteen neurons where each corresponds to a user. We use a supervised learning approach with back propagation algorithms, with a momentum of 0.2 and a learning rate of 0.3. Second, a Naive Bayes classifier Section. 3.3.4 and finally a Random Forest classifier is utilized (Section. 3.3.4). Similarly, for the verification process, the same aforementioned three classifiers are utilized, including a MLP NN that consists of 26 input neurons connected

to hidden layers of 14 neurons. The output layers have only two output neurons which are connected to the hidden layers. We use a supervised learning approach with back propagation algorithms, with a momentum of 0.2 and a learning rate of 0.3. Our analysis was performed with the support of the Weka machine learning tool [74].

6.2 Method

6.2.1 Participants

Fourteen users volunteered to participate in this experiment. There were thirteen males and one female, of ages ranging from 25 to 35. The subjects were briefed about the procedure and were given the choice to withdraw whenever they felt was necessary. None of them withdrew.

6.2.2 Apparatus

The incorporation of haptics to interpersonal communication system (developed in DISCOVER Lab) is done with the use of a depth camera, namely "ZCam"^{TM} [1] that captures a 2.5D scene consisting of color images and gray-scaled depth images that contain per pixel depth (Depth Image-based Haptic Representation (DIBHR)). The DIBHR representation is 2.5D because it represents incomplete 3D geometrical information that describes the visible scene, only visible to the user from the camera view. Therefore a user can't touch the part of the scene that is not visible, only the visible scenes are touchable. Moreover, DIBHR includes haptic properties and heightmap images. Haptic properties consist of stiffness, static friction, and dynamic friction, while heightmap images contain roughness of the video contents [16][17]

The 2.5D image captured by the ZCam^{TM} [1] camera is transformed into 3D space through the gray-scale depth image, and mapped as a texture (320x240 depth image). In order to have a fast rendering with no lags, the depth image is down-sampled. The haptic renderer observes the 3D interaction position and calculates the force between the user's hand position with the use of the Phantom OMNI device [68]and the transformed depth image. The calculation results guide the haptic device to provide proper force feedback, which lets the user feel the force of the interaction.[16].

6.2.3 Procedure

As mentioned previously, the purpose of our experiment is to continuously authenticate a user while s/he is performing a given task in a haptic application. Therefore, we asked users to touch a live 3-D representation reflected in the 3-D haptic environment. We believe this task can be considered as a random behaviour task. We assume this to be a free movement task since there were no criteria imposed on the type of movement. Users could touch, poke or rub any part of the 3-D live representation of their reflection. They could touch or poke their eyes, or open their mouth and touch their tongue, and so on. The number of possible users actions in such environment is extremely high.

As a first step in our conducted experiment, we explained about the system, the experiment procedure and the intended goal. Afterwards, we asked the users to try the system by freely gripping the haptic device and freely touching their 3-D representation, reflected in the 3-D virtual environment, until they felt comfortable with the use of the haptic device. Some subjects had been previously exposed to a haptic environment. They therefore familiarized themselves with the system in a short period of time, and were ready to start the experiment. Once the subjects were ready to perform the experiment, we asked them to freely touch the live 3-D representation of their reflection for 60 seconds. We repeated the same experiment twelve different times. After each trial, the subject reported that the session of 60 seconds of free movement is finished, and the haptic data is stored. Moreover, between every trial they had a break to relax in order to avoid haptic fatigue as described in [39].

In our experiment, the trials were divided into two separate data sets. The first six trials (t_1 to t_6) were used for training purposes, while a second data set (Hold-Out) represented by the last six trials (t_7 to t_{12}), were reserved for testing purposes.

6.3 Haptic Enabled Interpersonal Communication System Experimental Results

The final piece of our research puzzle is represented in the haptic-enabled interpersonal communication system experiment (Chapter 6). The target is to examine the feasibility of continuous user authentication during user interaction with a remote party. In this ex-

periment, we follow Approach 2 (Section 3.5.2) without considering the unbalanced data set problem as the user data is already balanced as a restriction enforced in data acquisition, and we use MLP Neural Networks (Section 3.3.4), Random Forest (Section 3.3.4), and Naive Bayes (Section 3.3.4) classifiers.

6.3.1 Identification

The goal of the user identification process is to reveal the identity of a given user. Therefore, in our algorithm (Section 3.5.2) a user is identified as the user with the highest $p(u_i|t, L)$. It is observed in Fig. 6.5 that users u_2, u_3, u_5, u_6, u_9, u_{12}, u_{14} are identified perfectly in all six trials ($t_7..t_{12}$) based on MLP NN algorithm, when considering a time-frame of 60 seconds. The remaining users have classification results that vary from 83% to 16% . Fig. 6.6 when considering a time-frame of 60 seconds and when Neural Networks 3.3.4 classification algorithm depicts an identification success rate of 79.7%. Moreover, Random Forest achieved classification results as high as 83.3%. In addition, the decrease of the selected time-frame (from 60 s to 5 s) resulted in a slight decrease in identification success rate, down to 75.5% and 78.8% for Neural Networks and Random Forest respectively. The advantage of continuously confirming a users identity in a short period of time (e.g. every 5 seconds) rather than a longer time-frame (e.g. 60 seconds) outweighs the slight decrease in classification performance. It makes the systems response to the user immediate, and in case of intrusion, in less than 5 seconds the legitimate user will be alerted. Moreover, it increases user security in a haptic-enabled interpersonal communication system where a single touch initiated by an illegitimate user may be interpreted in a wrong way, which might lead to inappropriate and embarrassing situations for the users. Naive Bayes classification performance was significantly low in comparison to NN and Random Forest.

6.3.2 Verification

When two remote users are communicating through a haptic enabled interpersonal communication system and exchange haptics touches, the system verifies the identity of each party continuously. Let's assume two users, Bob and Alice, are at two remote ends of a haptic-enabled interpersonal communication system. They identify themselves prior to using the system with the standard login for the haptic enabled interpersonal communi-

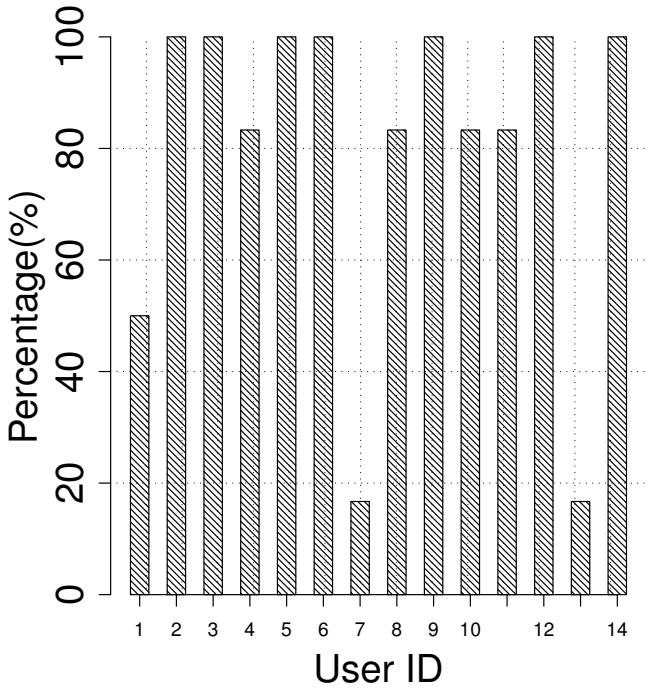


Figure 6.5: Average User Identification Rate for all trials t_7.. t_{12} on users u_1.. u_{14} (Experiment. 6

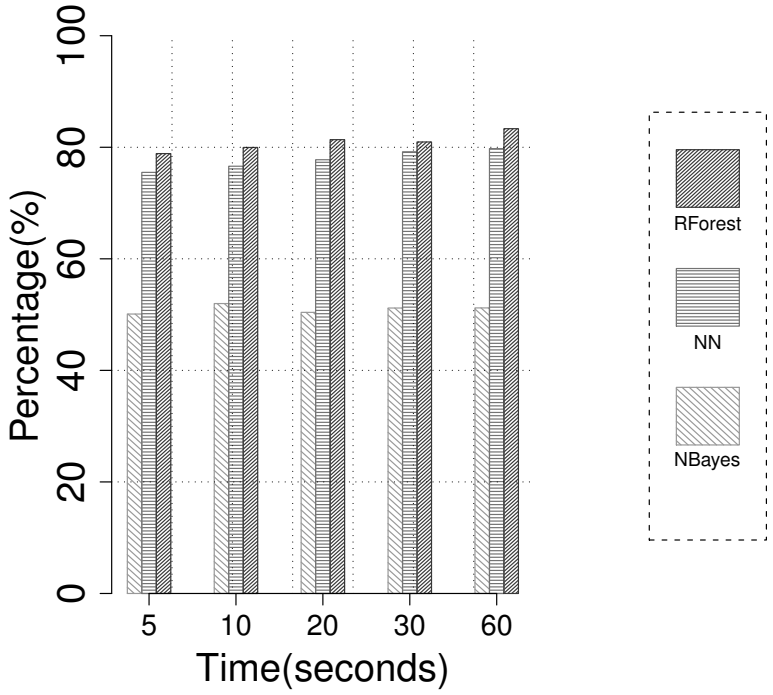


Figure 6.6: Overall average User identification rate for all Users u_1.. u_{14} on all testing trials t_7.. t_{12} while performing the identification process in every 5, 10, 20, 30 seconds, and a complete trial (60) while utilizing naive bayes, neural networks and random forest classifiers(Experiment. 6

Table 6.1: User verification results shows the FAR and FRR when a user-dependent threshold is utilized and adjusted based on trials t_7.. t_9. (Experiment. 6

Classifier	NN		RandomForest		NaiveBayes	
Time	FAR	FRR	FAR	FRR	FAR	FRR
5s	5.57%	17.71%	4.43%	16.57%	35.14%	3.50%
10s	6.0%	16.85%	5.07%	15.07%	31.14%	3.21%
20s	6.0%	12.28%	4.36%	14.64%	30.21%	2.36%
30s	5.14%	11.28%	3.86%	14.21%	28.29%	1.79%
60s	6.0%	11.92%	3.14%	11.93%	19.21%	10.71%

cation system. As soon as Bob or Alice initiate haptic movements or touches, the verification process continuously verifies the identity of the claimed users. The experimental results are based on the verification of data for fourteen users, as shown in Table 6.2, a False Acceptance Rate (FAR) of 2% and FRR of 9.5%, (FAR) of 1.92% @11.93% FRR is achieved using Neural Networks (NN), and Random Forest respectively, with a 60 second time-frame. Table 6.1 shows the results after considering (t_7..t_{12}) to adjust for the user-dependent threshold. An increase in FAR and FRR is observed as the time-frame decreases (down to FAR of 3.29% @ 14.93% FRR, and FAR of 4.43% @ 16.57% FRR for NN and Random Forest respectively) when a user is verified every 5 seconds. Table 6.1 illustrates the identity verification results when (t_7..t_9) are considered to adjust for the user-dependent threshold. 6% FAR @ 11.93% FRR and 3.14% FAR @ 11.93% FRR is achieved when a 60 seconds time-frame is used. As the time-frame decreases , the FAR remained consistent for NN, while a gradual increase of FRR is observed. A gradual increase of FAR and FRR is observed when Random Forest is used. Moreover, Naive Bayes classification performance achieved a high FAR compared to NN and Random Forest and relatively comparable results in terms of FRR.

6.3.3 Conclusion

The third stage captures user's fully random movements while communicating with a remote user. The haptic features are utilized to continuously authenticate the legitimate user. In the application of haptic-enabled interpersonal communication system, identity verification is utilized most of the time. If possible, user identification might be utilized

Table 6.2: User verification results shows the FAR and FRR when a user-dependent threshold is utilized and adjusted based on trials t_7.. t_{12}. (Experiment. 6

Classifier	NN		RandomForest		NaiveBayes	
Time	FAR	FRR	FAR	FRR	FAR	FRR
5s	5.57%	17.71%	2.50%	16.57%	18.21%	10.86%
10s	6.0%	16.85%	2.00%	15.50%	19.14%	7.71%
20s	6.0%	12.28%	1.57%	13.50%	16.57%	7.14%
30s	5.14%	11.28%	1.79%	13.64%	13.86%	7.14%
60s	6.0%	11.92%	1.29%	11.93%	15.50%	2.43%

to investigate the attacker's identity. As a result of the third stage, a FAR of 1.29% @ 11.93%FRR has been achieved with the utilization of Random Forest classifier.

Chapter 7

Conclusion and Futurework

7.1 Conclusion

With the advancement in multimedia systems and the advent of haptics to interpersonal communication systems, where users can see, show, hear, tell, touch and be touched, mouse and keyboard are no longer the dominant input devices. Touch, speech and vision will soon be the main methods of human computer interaction. Moreover, as the interpersonal communication increases, the need for securing user authentication grows. Traditional authentication systems will no longer be sufficient to authenticate the remote-party in a communication channel as the legitimate person. Moreover, Single Sign-On (SSO) authentication implies a risk as the communication duration increases. Therefore, relying on continuous authentication schemes that meet the interpersonal communication demands is a necessity. The user's identity is verified continuously as long as the communication between parties is established.

In this research, we established our work in three main stages, divided by the dynamic of user's movements in the targeted experiment. At the first stage, we studied users handwritten signatures with haptic information. The user's target in this experiment was to sign a check, which can be considered as a pre-defined task. Repeating the same signature was the user's target for this experiment. In the second stage, we moved to a semi-random movement by asking the user to draw a complex picture. There were no restrictions imposed on how the user sketches the picture or whether details are omitted in one sketch and present in another. Finally, we conducted an experiment on totally random and free user movements, where the users were asked to touch a remote

party as the haptic-enabled interpersonal communication continues.

In order to achieve our goal, we considered all possible means of feature selection and feature generation techniques. The proposed framework for haptic-based user authentication follows two approaches. The first approach targets, in addition to classification, dimensionality reduction by finding a proper and representative subset of feature space that would decrease the overhead of utilizing all haptic information. Consequently, this research analyzes the haptic features significant towards user authentication. This approach considers evolutionary computing, or more specifically genetic programming. The resulted analytical function is utilized for classification and feature reduction. For feature reduction, we proposed two methods in addition to the classical usage of analytical functions, where one method allows the application developer to specify the degree of reduction. The second approach focuses on classification, without feature reduction.

In this research, we encountered the unbalanced data set problem. In order to overcome this problem, many solutions were applied and proposed, such as Random minority Over-Sampling, Random majority Over-Sampling, weighting factors, and the proposed fitness functions to guide the evolution towards solutions that mitigate the unbalanced data set problem.

As a result, the use of dimensionality reduction methods reduced the feature space in the pre-defined task, from 10,000 features to 532 and 100 features when the two proposed feature reduction methods were applied respectively (Section 3.4.3). Moreover, it enhanced the classification success rate up to 97.78% VR @ (0.80% FAR, 19.23% FRR) and 96.76% VR @ (1.28% FAR, 26.92% FRR) when the ROS data set and SVM classifier is utilized. Moreover, the use of GEP generated models solely resulted in a relatively small number of features from the original 10,000, down to 6.42 features while maintaining 89.97% VR @ (8.14% FAR, 32.60% FRR). The proposed two feature reduction algorithm combined the features of several analytical functions and have increased the verification success rate while maintaining a high dimensionality reduction in the feature space. As for the proposed fitness functions, it resulted in a decrease in FRR and in the number of operations in the generated analytical functions.

In the semi-random task, PROFECE achieved an identity verification rate of FAR 15.79% @4.79% FRR when the verification occurs at the end of the task and using

Neural Network and while applying the weighting factors. The performance decreased as the verification time threshold decreased to 10 seconds, to FAR 12.93% @45.71% FRR.

The utilization of PROFECE to support random and continuous tasks resulted in promising results, where an identification success rate of 83.3% is achieved when Random Forest is utilized while considering a 60 seconds identification time threshold. Moreover, as we decrease the identification time threshold to 5 seconds to check the identity, the identification success rate slightly decreases to around 78.8%. Similarly for identity verification, a FAR of 1.29% @ 11.93%FRR and FAR of 2.50% @ 16.57% FRR is achieved when 60 and 5 seconds verification time threshold are utilized respectively. The design of PROFECE for continuous and random tasks signals an intrusion warning to the user if suspicious activity has been noticed. It is the end-users choice to either suspend or continue the session.

7.2 Future Work

The problem of continuous user authentication based on haptic information is still in its infancy. Exploring the efficiency in terms of overhead and processing costs would be targeted in our future work. Moreover, as we explored the haptic modality for user authentication, and similar to the recent DARPA futuristic vision [35], the combination of other user features such as keystroke dynamics, mouse movements, voice recognition and face recognition, in addition to haptics types, would be our focus in the near future. Moreover, exploring additional classification and feature reduction algorithms taking processing cost and time into consideration would give a broader view of what algorithms to utilize and what would be the incurred costs.

Further exploration on models generated by genetic programming for classification and feature selection will be conducted, such as considering ensembles of models. Moreover, in this research we tried to solve the unbalanced data set problem by utilizing a single objective function. However, our further research will expand to include multi-objective fitness functions and their effects on the generated models and user verification. Moreover, exploring other evolutionary algorithms for the purpose of enhancing the classification results of user identification and verification based on haptic information will be addressed in the near future.

The utilization of genetic programming for identity verification based on haptic interpersonal communication will be studied, and finding the relevant haptic features that minimize the transmitted haptic information will be explored.

Bibliography

[1] Zcam. www.3dvsystems.com.

[2] *Electronic Technology Directions to the Year 2000, Proceedings.* IEEE, May 1995.

[3] Neuron. http://en.wikipedia.org/wiki/Neuron, 01 2011.

[4] Reachin Technologies ab. Reachin Display. http://www.reachin.se/products/.

[5] D.W. Aha, D. Kibler, and M.K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.

[6] M. Alhalabi, Osama, and Susumu Horiguchi. Tele-handshake: A cooperative shared haptic virtual environment. In *PROC. EUROHAPTICS 2001, EUROGRAPHICS*, pages 60–64. Press, 2001.

[7] F A. Alsulaiman, J. Cha, and A. El Saddik. User identification based on handwritten signatures with haptic information. In Manuel Ferre, editor, *Haptics: Perception, Devices and Scenarios*, volume 5024 of *Lecture Notes in Computer Science*, pages 114–121. Springer Berlin / Heidelberg, 2008.

[8] F. A. Alsulaiman, J. Cha, and A. El Saddik. User authentication using haptic information in drawing. Submitted to Springer Journal on Multimedia Tools and Applications. Springer MTAP., February 2012.

[9] F. A. Alsulaiman, Jongeun Cha, and A. El Saddik. Preliminary results of a user identification using haptic information. In *IEEE Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2008. VECIMS 2008.*, pages 10 –13, 2008.

[10] F. A. Alsulaiman, Sakr, J.J. N., Valdes, and El Saddik A. Identity verification based on handwritten signatures with haptic information using genetic programming. *ACM Transactions on Multimedia Computing, Communication and Applications (ACM TOMCCAP).*, 9(2), May 2013.

[11] U. Bhawan, M. Johnston, and M. Zhang. Differentiating between individual class performance in genetic programming fitness for classification with unbalanced data. In *In the Proceedings of the Eleventh Conference on Congress on Evolutionary Computation*, pages 2802–2809, 2009.

[12] C. Blacke and C. Merz. Uci repository of machine learning databases, 1998.

[13] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[14] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and regression trees.* Wadsworth International Group, 1984.

[15] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining Knowledge Discovery*, 2:121–167, 1998.

[16] J. Cha, M. Eid, A. Barghout, A.M Rahman, and A. El Saddik. Hugme: Synchronous haptic teleconferencing. In *ACM International Conference on Multimedia, The Multimedia Grand Challenge*, Beijing, China, Oct. 2009.

[17] J. Cha, M. Eid, and A. El Saddik. Touchable 3d video system. *ACM Transactions on Multimedia Computing Communications and Applications*, 5(4):1–25, 2009.

[18] J. Cha, M. Eid, L. Rahal, and A. El Saddik. Hugme: An interpersonal haptic communication system. In *IEEE International Workshop on Haptic Audio visual Environments and Games, 2008. HAVE 2008.*, pages 99 –102, 2008.

[19] C-C. Chang and C-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[20] N. V. Chawla, K. W. Bowyer, Lawrence O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321 – 357, 2002.

[21] T.M. Cover and P.E. Hart. Nearest neighbor pattern classification. *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, 13:21–27, 1967.

[22] C. DiSalvo, G. Francine, J. Forlizzi, and E. Montgomery. The hug: An exploration of robotic form for intimate communication. In *Procsedings of the 2003 iEEE intemalionai Workshop on Robot and Human interactive Communication*, pages 403–408. IEEE, 2003.

[23] J. Doucette and M.I. Heywood. Gp classification under imbalanced data sets: Active sub-sampling and auc approximation. In *In the Proceedings of the European Conference on Genetic Programming (EuroGP)*, 2008.

[24] J. Eggermont, A.E. Eiben, and J.I.V. Hermert. Adapting the fitness function in gp for data mining. In *In Proceedings of the European Conference on Genetic Programming (EuroGP)*, volume 1598 of LNCS, pages 193–202, 1999.

[25] M. Eid, M. Orozco, and A. El Saddik. A guided tour in haptic audio visual environment and applications. *Int. J. of Advanced Media and Communication*, 1(3):265 – 297, 2007.

[26] A. El Saddik. The potential of haptics technologies. *IEEE Instrumentation Measurement Magazine*, 10(1):10 –17, 2007.

[27] A. El Saddik, M. Orozco, Y. Asfaw, S. Shirmohammadi, and A. Adler. A novel biometric system for identification and verification of haptic users. *IEEE Transactions on Instrumentation and Measurement*, 56(3):895 –906, 2007.

[28] B. Eoff and T. Hammond. Who dotted that 'i'?: Context free user differentiation through pressure and tilt pen data. pages 149–156, Kelowna, British Columbia, Canada, May 2009.

[29] C. Ferreira. Gene expression programming: A new adaptive algorithm for problem solving. *Journal of Complex Systems*, 13(2):87–129, 2001.

[30] C. Ferreira. *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence*. Springer Verlag, Germany, 2006.

[31] H. Gamboa and A. Fred. An identity authentication system based on human computer interaction behaviour. In *Proc. 3rd Int. Workshop on Pattern Recognition in Information Systems*, pages 46–55, 2003.

[32] C. Gathercole and P. Ross. Dynamic training subset selection for supervised learning in genetic programming. *In Parallel Problem Solving in Nature III*, 866 of LNCS:312–321, 1994.

[33] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias / variance dilemma. *Neural Computation*, 4:1–58, 1992.

[34] M. Gtlein, E. Frank, M. Hall, and A. Karwath. Large-scale attribute selection using wrappers. In *In Proc IEEE Symposium on Computational Intelligence and Data Mining*, pages 332–339, 2009.

[35] Mr. Richard Guidorizzi. Active authentication, April 2012.

[36] I. Guyon, V. Vapnik, B. Boser, L. Bottou, and S.A. Solla. Structural risk minimization for character recognition. *Advances in Neural Information Processing Systems*, 4:471479, 1992.

[37] A. Haans and W. Ijsselsteijn. Mediated social touch: a review of current research and future directions. *Virtual Reality*, 9(2):149–159, 2006.

[38] M.A. Hall and G. Holmes. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, 15(3):1437–1447, May/June 2003.

[39] A. Hamam, N. D. Georganas, F. A. Alsulaiman, and A. El Saddik. Deducing user's fatigue from haptic data. In *Proc. of International Conference on Multimedia (MM '10)*, pages 1083–1086, New York, NY, USA, 2010. ACM.

[40] M.A. Hearst, S.T. Dumais, E. Osman, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, Jul/Aug 1998.

[41] J. V. Hulse, T. M. Khoshgoftaar, and Napolitano A. Experimental perspectives on learning from imbalanced data. In *in the Proceedings of the 24th International Conference on Machine Learning*, pages 935–942, 2007.

[42] D.R. Hush and B.G. Horne. Progress in supervised neural networks. *IEEE Signal Processing Magazine*, 10(1):8–39, January 1993.

[43] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intell. Data Anal.*, 6:429–449, October 2002.

[44] C-H. Jiang, S. Shieh, and J-C. Liu. Keystroke statistical learning model for web authentication. In *Proceedings of the 2nd ACM symposium on Information, computer and communications security*, ASIACCS '07, pages 359–361, New York, NY, USA, 2007. ACM.

[45] G. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proc. of 11th Conference on Uncertainty in Artificial Intelligence*, pages 338–345, 1995.

[46] R. Joyce and G. Gupta. Identity authentication based on keystroke latencies. *Commun. ACM*, 33:168–176, February 1990.

[47] J. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

[48] J. Koza. *Genetic programming II: Automatic discovery of reusable programs*. MIT Press, 1994.

[49] M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: one-sided selection. In *Proc. 14th International Conference on Machine Learning*, pages 179–186, 1997.

[50] J. Lee, S. Choi, and B. Moon. An evolutionary keystroke authentication based on ellipsoidal hypothesis space. In *Proc. 9th Annual Conf. Genetic and Evolutionary Computation (GECCO)*, pages 2090–2097, 2007.

[51] L.L. Lee, T. Berger, and E. Aviczer. Reliable online human signature verification systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(6):643 –647, June 1996.

[52] R.P. Lippmann. An introduction to computing with neural nets. *IEEE Acoustics. Speech and Signal Processing Magazine*, 4(2):4–22, 1987.

[53] S. Luke, L. Panait, G. Balan, S. Paus, Z. Skolicki, E. Popovici, J. Harrison, J. Bassett, R. Hubley, and A. Chircop. Ecj. a java-based evolution computing research system, 2007. Evolutionary Computation Laboratory, George Mason University.

[54] B. Malek, M. Orozco, and El Saddik A. Novel shoulder-surfing resistant haptic-based graphical password. In *Proceedings of the Eurohaptics 2006 conference*, July 2006.

[55] F. Monrose, M. K. Reiter, and S. Wetzel. Password hardening based on keystroke dynamics. In *Proceedings of the 6th ACM conference on Computer and communications security*, CCS '99, pages 73–82, New York, NY, USA, 1999. ACM.

[56] F. Monrose and A. Rubin. Authentication via keystroke dynamics. In *Proceedings of the 4th ACM conference on Computer and communications security*, CCS '97, pages 48–56, New York, NY, USA, 1997. ACM.

[57] D.C. Montgomery and E.A. Peck. *Introduction to Linear Regression Analysis.* John Wiley and Sons, Inc.,, 2nd edition edition, 1992.

[58] F. Mueller, F. Vetere, M. R. Gibbs, J. Kjeldskov, S. Pedell, and S. Howard. Hug over a distance. In *CHI '05 extended abstracts on Human factors in computing systems*, CHI '05, pages 1673–1676, New York, NY, USA, 2005. ACM.

[59] M. Orozco, Y. Asfaw, A. Adler, S. Shirmohammadi, and A. El Saddik. Identification of participant in haptic systems. In . *Instrumentation and Measurement Technology Conference*, 2005.

[60] M. Orozco, Y. Asfaw, S. Shirmohammadi, A. Adler, and A. El Saddik. Haptic-based biometrics: A feasibility study. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2006 14th Symposium on*, pages 265 – 271, 2006.

[61] M. Orozco and A. El Saddik. Recognizing and quantifying human movement patterns through haptic-based applications. In *Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2005. VECIMS 2005. Proceedings of the 2005 IEEE International Conference on*, page 5 pp., 2005.

[62] M. Orozco, M. Graydon, S. Shirmohammadi, and A. A. El Saddik. Experiments in haptic-based authentication of humans. *Journal of Multimedia Tools and Applications*, 37(1):73–92, 2008.

[63] M. Orozco, B. Malek, M. Eid, and A. El Saddik. Haptic-based sensible graphical password. In *In Proc. Virtual Concept 2006*, Playa Del Carmen, Mexico, November 2006.

[64] G. Patterson and M. Zhang. Fitness functions in genetic programming for classification with unbalanced data. In *In Proceedings of the 20th Australian Joint Conference on Artificial Intelligence.*, 2007.

[65] M. Pazzani, C. Merz, P. Murphy, K. Ali, Hume T., and C. Brunk. Reducing misclassification costs. In *Proc. 11th Int. Conf. of Machine Learning*, pages 217–225, 1994.

[66] M. Pusara and C. E. Brodley. User re-authentication via mouse movements. In *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, VizSEC/DMSEC '04, pages 1–8, New York, NY, USA, 2004. ACM.

[67] F. Rosenblatt. The percptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.

[68] SensAble Technologies. Sensable technologies homepage, 2010.

[69] D. Song, M.I. Heywood, and A.N. Zincir-Heywood. Training genetic programming on half a million patterns: An example from anomaly detection. *IEEE Transactions on Evolutionary Computation*, 9(3):225–239, 2005.

[70] K. Sung. *Learning and Example Selection for Object and Pattern Recognition.* PhD thesis, MIT, 1996.

[71] H. TAO. Graphical password authentication arrangement for computer, displays grid with multiple horizontal and vertical lines, in response to user request for accessing restricted resource, using which user enters password, 2005.

[72] I. Tomek. Two modifications of cnn. *Systems, Man and Cybernetics, IEEE Transactions on*, 6(11):769 –772, 1976.

[73] J.J. Valdés, A.J. Barton, and R. Orchard. Exploring medical data using visual spaces with genetic programming and implicit functional mappings. In *Proc. of Genetic and Evolutionary Computation*, pages 2953–2960, 2007.

[74] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, SF, CA, 2nd edition edition, 2005.

[75] L. Yan, R. Dodier, M.C. Mozer, and R. Wolniewicz. Optimizing classifier performance via the wilcoxon-mann-whitney statistic. In *In Proceedings of the International Conference on Machine Learning*, pages 848–855, 2003.