Summer 2014

Novel rejection methods and fusion approaches for multi-biometric verification

Md Shafaeat Hossain

NOVEL REJECTION METHODS AND FUSION APPROACHES FOR MULTI-BIOMETRIC VERIFICATION

by

Md Shafaeat Hossain, B.Sc., M.S., M.S., M.S.

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

COLLEGE OF ENGINEERING AND SCIENCE
LOUISIANA TECH UNIVERSITY

August 2014

LOUISIANA TECH UNIVERSITY

THE GRADUATE SCHOOL

May 07, 2014
Date

We hereby recommend that the dissertation prepared under our supervision

by Md Shafaeat Hossain

entitled

Novel Rejection Methods and Fusion Approaches for Multi-biometric
Verification

be accepted in partial fulfillment of the requirements for the Degree of
Doctor of Philosophy

Vir Virander Phoha
Supervisor of Dissertation Research

Weizhong Dai
Head of Department

Computational Analysis and Modeling
Department

Recommendation concurred in:

Jinko Kanno

Weizhong Dai

Advisory Committee

Approved:
Director of Graduate Studies

Approved:
Sheryl S. Shoemaker
Dean of the Graduate School

Hisham Hegab
Dean of the College

ABSTRACT

This dissertation proposes methods and algorithms to improve the performance of biometric verification systems. It introduces a new rejection method, "symmetric rejection method," for multi-stage biometric verification. The *symmetric rejection method* significantly improves the performance over the state of the art rejection methods and controls the genuine reject rate which has not been specifically addressed in earlier studies. The dissertation also proposes a new fusion framework for multi-biometric verification systems, which achieves accuracy higher than parallel fusion framework, and provides convenience to genuine users. In addition, it proposes a framework consisting of impostor score based normalization, impostor score based rejection, and fusion to lower the verification errors of continuous keystroke verification with weak templates. It introduces a new formulation to incorporate the *reject option* in verification with weak templates and develops a new impostor score based rejection method called "Order Statistic rejection method". Results show that the proposed framework in conjunction with the *Order Statistic rejection method* significantly reduces the equal error rates of continuous keystroke verification with weak templates.

APPROVAL FOR SCHOLARLY DISSEMINATION

The author grants to the Prescott Memorial Library of Louisiana Tech University the right to reproduce, by appropriate methods, upon request, any or all portions of this Dissertation. It is understood that "proper request" consists of the agreement, on the part of the requesting party, that said reproduction is for his personal use and that subsequent reproduction will not occur without written approval of the author of this Dissertation. Further, any portions of the Dissertation used in books, papers, and other works must be appropriately referenced to this Dissertation.

Finally, the author of this Dissertation reserves the right to publish freely, in the literature, at any time, any or all portions of this Dissertation.

Author Md. Shafaeat H.

Date 07/16/2014

DEDICATION

I dedicate my doctoral dissertation to my parents Abdul Jalil and Sufia Begum, my sisters Zubyda Pervin, Zulekha Akter, and Jesmin Sultana, my late grandma Lal Banu, my loving wife Tasnima Jannat, and last but not least, my sweet daughter, Fatimah Hossain.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Dr. Vir Phoha for the continuous support of my Ph.D study and research. Besides my advisor, I would like to thank Dr. Kiran Balagani and Dr. Enamul Karim for their guidance, encouragement, and insightful comments. Specially, Dr. Balagani's guidance helped me throughout my research.

My sincere thanks also goes to the rest of my dissertation committee: Dr. Rastko Selmic, Dr. Weizhong Dai, and Dr. Jinko Kanno for their time and effort.

Last but not the least, I would like to thank all of my family members for supporting me spiritually throughout my life.

CHAPTER 1

INTRODUCTION

1.1 Overview

A biometric verification system uses physiological traits (for example, fingerprint, face, iris, hand-geometry, retina, etc.) and/or behavioral traits (for example, keystroke dynamics, mouse movement, signature, speech, etc.) of an individual to verify his/her identity claim. A unibiometric verification system that uses a single biometric trait suffers from several practical problems such as noisy sensor data, non-universality and lack of uniqueness of biometric traits, and spoof attacks (see [1], [2]). As a result, a unibiometric verification system fails to meet the tight requirement of real-world applications. A multi-biometric verification system seeks to alleviate these problems by fusing information from multiple biometric sources (see [3], [4], [5], [6]). These sources may be multiple biometric traits (for example, face and fingerprint), multiple instances of the same biometric (for example, left index fingerprint and right index fingerprint of a person), multiple matching algorithms for the same biometric (for example, two different face matchers: principal component analysis based matcher and linear discriminant analysis based matcher), or multiple sensors for the same biometric (for example, optical and ultrasonic fingerprint sensor) (see [1] for details). Several studies show that fusion of information from multiple biometric sources significantly lowers the verification errors (see [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]).

Fusion of information can be performed in two different modes: (1) parallel mode and (2) serial mode. In parallel mode (see [1], [4], [6], [9], [18], [19], [20], [21]), to verify a user U, an n-biometric verification system collects n biometric traits from U, processes each trait individually, combines the information, and gives verification decision on the combined information. In contrast, in serial mode (see [22], [23], [24], [25], [26], [27], [28]), to verify a user U, an n-biometric verification system collects the first biometric trait in the processing chain from U, process it, and gives verification decision on the processed information if it has enough evidence to classify U as genuine or an impostor. If the verification system is not confident enough to ascertain whether U is genuine or an impostor, it *rejects* the sample and collects the sample of the next biometric trait to get more evidence for classification. The verification system collects the n^{th} biometric trait only when it fails to give the verification decision using biometric traits 1 through $n-1$.

A serial fusion based biometric verification system is referred to as a *multi-stage* biometric verification system. The option to reject the 'confusing' samples in stages 1 through $n-1$ of an n-stage biometric verification system is called *reject option* (see [29], [30], [31], [32], [33], [34], and [35]), which builds the skeleton of the multi-stage biometric verification system. Reject option is exercised by selecting a reject region that says which samples are to be rejected. A sample is rejected if the corresponding matching score falls inside the reject region. A rejection method is used to select an appropriate reject region.

In this dissertation, we propose a new rejection method, referred to as *symmetric rejection method*, for multi-stage biometric verification. We empirically show that the symmetric rejection method significantly improves the performance over the state of the art rejection methods. Compared to the existing rejection methods, the symmetric rejection method has two notable advantages: (1) it allows to control the genuine reject rate–the proportion of genuine scores that are erroneously falling inside the reject region, which has not been specifically addressed in earlier studies, and (2) it allows to compute the reject region without estimating the underlying probability density function of the base scores.

We also propose an optimized fusion framework for multi-biometric verification systems, which combines the advantages of parallel fusion and serial fusion. Both parallel fusion and serial fusion involve multiple biometric verifiers and/or multiple biometric traits. Between the two modes of fusion, parallel fusion has received more attention from researchers because of its higher accuracy [2]. However, several recent studies such as [23], [24], [26], and [27] have questioned the applicability of parallel fusion in applications that involve a large population of users or a great number of biometric transactions because parallel fusion has a much longer verification time and thereby causes inconvenience to genuine users. Serial fusion provides convenience to genuine users by allowing them to submit a subset of biometric traits.

Unfortunately, studies such as [26] and [36] show that serial fusion cannot reach the accuracy level of parallel fusion and hence its applicability is questionable in high security applications that require robustness to forgeries, robustness to enrollment problems, etc. In this work, we propose a fusion framework which (1) achieves an accuracy level higher than parallel fusion, and at the same time (2) provides convenience to genuine users. Hence, our proposed fusion framework is applicable to high security applications as well as applications that involve a large population of users or a huge number of biometric transactions.

We also propose a framework for continuous keystroke verification with "weak" templates. In continuous keystroke verification, a user's template is built from keystroke samples collected in multiple enrollment sessions. Several studies in keystroke verification ([37], [38], and [39]) show that increasing the number of enrollment samples with "multi-session enrollment" lowers the verification error rates. However, a practical limitation with "multi-session enrollment" is that it assumes users are either available or solicitable multiple times. Though this assumption is usually true in a controlled laboratory setting, its validity is questionable in realistic cyber environments. For example, in cloud computing environments, thousands of users access system resources remotely (over the network) and it is unrealistic to expect that every user is available in a timely fashion to submit enrollment samples. Some users may even access the system so infrequently that repetitive collection of enrollment samples is impractical. With such users, a keystroke verification system is left to use a "weak" template, which is a template built from insufficient enrollment samples (*e.g.*, enrollment samples collected in a single session).

The goal of this work is to minimize the equal error rates (EERs) of continuous keystroke verification when "weak" templates are used. In this regard, we propose a framework comprising of impostor score based normalization, impostor score based rejection, and fusion. We introduce a new formulation to incorporate the reject option in verification with weak templates and develop a new impostor score based rejection method called Order Statistic (OS) rejection method. We compare the performance of OS rejection method with two other impostor score based rejection methods–(1) Otsu [40] and (2) Gaussian. Experimental results show that our proposed framework significantly reduces the EERs of continuous keystroke verification with weak templates and the OS rejection method achieves better error-reject trade-off than Otsu and Gaussian rejection methods.

1.2 Our Contributions

In this dissertation, our focus is to improve the performance of biometric verification systems, and at the same time, provide significant convenience to genuine users. Below, we briefly describe our contributions.

1. We propose a new rejection method called *symmetric rejection method* for multi-stage biometric verification. Compared to existing rejection methods, symmetric rejection method has the following advantages:
   - Compared to the rejection methods proposed by [23], [24], and [25] that reject all the genuine scores inside the confusion region (which can considerably increase user inconvenience, especially when the volume of biometric transactions is high), the symmetric rejection method allows the administrator to control the genuine reject rate in each stage of a multi-stage verification system.
   - Compared to the rejection methods proposed by [26], [27], and [28] that require estimation of probability density function of the scores (which can

turn out to be difficult and expensive), the symmetric rejection method allows us to calculate reject region directly from the scores.

2. We analytically show how the symmetric rejection method reduces false alarm rate and impostor pass rate when it is used in a multi-stage biometric verification system. We validate our theory by experimenting on a four-stage verification system. We perform experiments using NIST multi-modal database [41], which consists of verification scores originated from four individual verifiers. Our results show that the symmetric rejection method significantly reduces false alarm rate and impostor pass rate. Using the symmetric rejection method, we achieve (1) a minimum false alarm rate of 0.0039, which is 91.58 percent less than the equal error rate of the top performing individual verifier and (2) a minimum impostor pass rate of 0.0203, which is 56.16 percent less than the equal error rate of the top performing individual verifier.

3. We compare the performance of the symmetric rejection method with two existing rejection methods: (1) SPRT-based method [26] and (2) Marcialis *et al.*'s method [23]. Experimental results show that to achieve the same value of area under ROC curve (AUC), genuine users require less number of stages with the symmetric rejection method compared to SPRT-based and Marcialis *et al.*'s rejection methods. This indicates that the symmetric rejection method can provide better user convenience, which is a desirable attribute, especially for applications involving a large population of users or a great number of biometric transactions.

4. We propose a new fusion framework for multi-biometric verification systems, which (1) achieves an accuracy level higher than parallel fusion, and at the same time (2) provides a significant amount of convenience to genuine users. In other words, we propose a fusion framework which combines the advantages of parallel fusion and serial fusion.

5. We theoretically show that the false alarm rate and impostor pass rate obtained by our proposed fusion framework is less than or equal to the false alarm rate and impostor pass rate obtained by parallel fusion. We validate our theory by experimenting on ***two*** multi-biometric verification systems, where each verification system uses three different biometric traits. We use NIST multimodal database [41] in our experiments. The experimental results provide a considerable amount of evidence that the proposed fusion framework improves the performance over the parallel fusion framework, and at the same time, provides a significant amount of convenience to the genuine users. Specifically:

   - With one verification system, (1) we achieve a minimum EER of 0.4561 percent, which is 13.63 percent less than the EER obtained with parallel fusion and 89.29 percent less than the EER obtained with the top performing individual verifier, and at the same time, (2) we achieve the following user convenience: 77.29 percent genuine users received verification decision by using only one biometric trait, *i.e.*, 77.29 percent genuine users did not need to submit second or third biometric traits and 7.36 percent genuine users received verification decision by using only two biometric traits, *i.e.*, 84.65 percent genuine users received verification decisions without submitting the third biometric trait. In contrast, in parallel fusion, every user had to submit three biometric traits.

- With another verification system, (1) we achieve a minimum EER of 0.3399 percent, which is 3.08 percent less than the EER obtained with parallel fusion and 92.02 percent less than the EER obtained with the top performing individual verifier, and at the same time, (2) we achieve the following user convenience: 25.89 percent genuine users received verification decision by using only one biometric trait, *i.e.*, 25.89 percent genuine users did not need to submit second or third biometric traits, and 57.29 percent genuine users received verification decision by using only two biometric traits, *i.e.*, 83.18 percent genuine users received verification decisions without submitting the third biometric trait. In contrast, in parallel fusion, every user had to submit three biometric traits.

6. We propose a framework comprised of *impostor score based normalization*, *impostor score based rejection*, and *fusion* to lower the EERs of continuous keystroke verification with weak templates.

7. We introduce a *new formulation* to incorporate the reject option in verification with weak templates and develop a new *impostor score based* rejection method called the Order Statistic (OS) rejection method. Furthermore, we adapt: 1) the Otsu threshold selection method [40] and 2) the Gaussian assumption of scores to our rejection formulation and study how they perform as *impostor score based* rejection methods.

8. By conducting experiments on a large keystroke database of 1100 users, we show that all three rejection methods significantly reduce the EERs (*i.e.*, our rejection formulation has considerable impact on reducing the EERs of continuous keystroke verification with weak templates). We compare the performance of the OS rejection method with Otsu rejection method (non-parametric) and Gaussian rejection method (parametric). Results show that OS rejection outperforms both Otsu and Gaussian in terms of error-reject trade-off. We achieve 59.97 percent to 86.74 percent *reduction* in average EERs compared to the individual verifiers when we use OS rejection method in conjunction with impostor score based normalization and fusion.
9. We show that impostor score based normalization and fusion significantly reduce the EERs of continuous keystroke verification with weak templates. Though impostor score based normalization and fusion were previously studied with several biometric modalities, to the best of our knowledge, this is the first work to study its performance in a continuous keystroke verification setting. With impostor score based normalization and fusion, we achieve 47.47 to 69.67 percent reduction in average EERs compared to the individual verifiers.

1.3 Organization of the Dissertation

In Chapter 2, we briefly describe (1) fusion in biometric verification, (2) the basics of reject option–which plays a key role in multi-stage biometric verification, (3) biometric verification rule with reject option, and (4) the basics of continuous keystroke verification. In Chapter 3, we discuss related research and our motivation.

In Chapter 4, we introduce the symmetric rejection method, analyze its performance, give empirical validation, and compare its performance with SPRT-based [26] and Marcialis *et al.*'s [23] rejection methods. In Chapter 5, we introduce the proposed fusion framework by theoretically showing how it (1) gives better performance than parallel fusion and (2) provides convenience to genuine users, and empirically validate our theory. In Chapter 6, we describe the proposed framework of continuous keystroke verification with weak templates, introduce OS, Otsu, and Gaussian rejection methods, give experimental details, and analyze the results. We conclude and give future directions in Chapter 7.

CHAPTER 2

BACKGROUND

2.1 Fusion of Information in Biometric Verification

Fusion in a multi-biometric verification system can be done at four different levels of information: (1) sensor level, (2) feature level, (3) matching score level, and (4) decision level. Score level fusion is generally preferred because matching scores contain sufficient information to distinguish genuine users and impostors, and at the same time, they are relatively easy to obtain (see [9], [10], [42], [43]). Given a number of unimodal biometric verification systems, it is possible to generate matching scores for a specified number of users without having any knowledge of the underlying feature extraction methods and matching algorithms of the unimodal verification systems. Hence, combining information using score level fusion is feasible and practical (see [10], [43]).

Parallel fusion and serial fusion are two widely used modes of information fusion. In parallel mode (see [1], [4], [6], [9], [18] [19], [20], [21]), to verify a user U, an n-biometric verification system collects n biometric traits from U, processes each trait individually, combines the information, and gives verification decision on the combined information. In Section 3.2, we discuss different approaches to parallel fusion proposed in the literature.

In serial mode (see [22], [23], [24], [25], [26], [27], [28]), to verify a user U, an n-biometric verification system collects the first biometric trait in the processing chain from U, processes it, and gives verification decision on the processed information if it has enough evidence to classify U as genuine or an impostor. If the verification system is not confident enough to ascertain whether U is genuine or an impostor, it *rejects* the sample and collects the sample of the next biometric trait to get more evidence for classification. The verification system collects the n^{th} biometric trait only when it fails to give verification decision using biometric traits 1 through $n-1$. In Section 3.2, we discuss different approaches to serial fusion proposed in the literature.

A serial fusion based biometric verification system is referred to as the *multi-stage* biometric verification system. The option to reject the 'confusing' samples in stages 1 through $n-1$ of an n-stage biometric verification system is called *reject option* (see [29], [30], [31], [32], [33], [34], and [35]), which builds the skeleton of the multi-stage biometric verification system. In the following subsections, we describe the reject option basics and biometric verification rule with reject option.

2.2 Reject Option Basics

In classification, a test pattern S is classified into one of r classes $\{\omega_1, \cdots, \omega_r\}$. The goal of a rejection rule is to improve classification accuracy by allowing the classifier to *not classify* S if the classifier has low confidence on its decision on S. *Ambiguity-reject* [29] and *distance-reject* [44] are two widely used rejection rules. Let $P(\omega|S)$ be the posterior probability of a class given S. S is assigned to ω_* if a) $P(\omega_*|S) = \max\{P(\omega_1|S), \cdots, P(\omega_r|S)\}$, b) $\frac{P(\omega_*|S)}{P(\omega_j|S)} \geq t_1$, $j = 1, \cdots, r$, $\omega_j \neq \omega_*$, and

c) $P(\omega_*|S) \geq t_2$. Here, ω_* is one of the r classes, a) is Bayes classification rule, b) is ambiguity-reject rule, c) is distance-reject rule, and t_1 and t_2 are thresholds.

Ambiguity-reject assumes that all classes are known apriori, test patterns belong to one of the known classes, and all classes are well represented in the training data. Under these assumptions, ambiguity-reject rejects patterns that occur in the overlapping regions between classes. On the other hand, distance-reject assumes that a test pattern may belong to a class unknown to the classifier. Under this assumption, it rejects patterns that are "distant" from the known classes. The performance of a rejection rule is analyzed using an error-reject trade-off curve, which shows how the classification error decreases as more test patterns are rejected.

2.3 Reject Option in Multi-stage Biometric Verification

Reject option plays a key role in multi-stage biometric verification. Specifically, in an n-stage biometric verification system (*e.g.*, [22], [23], [24], [25], [26], [27], [28], and [45]), if the verifier in stage i is not confident enough to decide whether the sample is genuine or an impostor, the sample is *rejected* and a new sample is submitted to the verifier in stage $i + 1$ to get a more confident decision. If all the verifiers in stages 1 through $n - 1$ fail to give a genuine or impostor decision, the verifier in stage n (last stage) gives the final decision. Below, we formally state the biometric verification rule with the reject option.

Biometric Verification Rule with Reject Option: Let s denote a verification score output by verifier v when a biometric sample S is matched with a claimed template C. Assuming v outputs a dissimilarity score (*e.g.*, Euclidean distance between features of S and C), biometric verification with the reject option is implemented as:

$$S \text{ is } \textit{genuine} \text{ if } s < L, \; \textit{impostor} \text{ if } s > R, \text{ else } \textit{reject } S, \tag{2.1}$$

where $[L, R]$ is the reject region defined by thresholds L and R. In Figure 2.1, we illustrate the reject region $[L, R]$ ('LR' hereafter). Here we assume X and Y are genuine and impostor score sets, respectively, generated by the verifier v for user U; $f_G(x)$ and $f_I(y)$ are the distributions estimated from scores in X and Y, respectively. Without loss of generality, we assume verifier v outputs *real-valued dissimilarity* scores. In the scoreline $[Z, O]$, ZE_2 is the genuine score region, E_1O is the impostor score region, and E_1E_2 is the confusion region where $f_G(x)$ and $f_I(y)$ overlap.

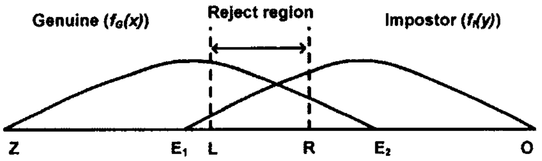


Figure 2.1: Genuine score distribution $f_G(x)$ and impostor score distribution $f_I(y)$ along with reject region LR, genuine score region ZE_2, impostor score region E_1O, and confusion region E_1E_2.

Because scoreline $[Z, O]$ (see Figure 2.1) is a real line, there are potentially an infinite number of regions that verification rule in (2.1) can use as a reject region. Depending upon the location and width, different reject regions can yield different error rates. There are two types of error rates associated with the verification rule in (2.1):

$$FAR^r = \frac{\#\text{ of genuine scores declared as impostor}}{\text{Total } \#\text{ of genuine scores} - \#\text{ of genuine scores in } LR} \tag{2.2}$$

and

$$IPR^r = \frac{\text{\# of impostor scores declared as genuine}}{\text{Total \# of impostor scores} - \text{\# of impostor scores in } LR}, \tag{2.3}$$

where FAR^r is the false alarm rate, IPR^r is the impostor pass rate, and LR is the reject region. We use superscript 'r' to specify that the false alarm rate and impostor pass rate are obtained by exercising *'reject option'*.

Obviously, the target of any verification rule is to minimize the error rates. However, when the reject option is added to a verification rule, one more thing needs to be considered: how many scores are being rejected by the reject region? There are two types of reject rates associated with the verification rule in (2.1):

$$GRR = \frac{\text{\# of genuine scores in } LR}{\text{Total \# of genuine scores}} \tag{2.4}$$

and

$$IRR = \frac{\text{\# of impostor scores in } LR}{\text{Total \# of impostor scores}}, \tag{2.5}$$

where GRR is the genuine (score) reject rate, IRR is the impostor (score) reject rate, and LR is the reject region. Note that, while rejecting impostor scores incurs no cost in terms of user inconvenience, erroneously rejecting genuine scores translates to both user and administrator inconvenience. The problem becomes severe in the applications that involve a large population of users or a great number of biometric transactions, for example, e-commerce, ATMs, etc. In such applications, selecting an *inappropriate* reject region can result in a high genuine reject rate and thereby render the verifier impractical. Therefore, it is necessary, in such applications, to control the genuine reject rate (GRR).

In summary, we expect a reject region should be chosen such that: (1) it reduces false alarm rate and impostor pass rate, and at the same time, (2) it maintains a specified genuine reject rate.

2.4 Basics of Continuous Keystroke Verification

Continuous keystroke verification (see [46], [47]) is implemented in two phases: the *training* phase and the *testing* phase. In the training phase, we extract keystroke features from a user's typing sample (enrollment text), remove outliers, and generate templates. In the testing phase, we use a verifier to match test attempts (extracted from verification text) to the user's template and output verification scores. In continuous verification, extracting attempts from verification text and matching them against the user's template is a continuous process.

We experimented with three widely used keystroke features: 1) key interval latency (the time between release of a key and press of the next key), 2) key hold latency (the time between press and release of a key), and 3) key press latency (the time between press of a key and press of the next key). A user's template is a 26-by-26 matrix in which each cell corresponds to an English digraph (*i.e.* aa, ab, ac, ..., zy, zz). Each digraph can occur multiple times in the user's sample. Therefore, each cell stores multiple latency values and their aggregate statistics (mean and standard deviation). Typically, a key hold latency corresponds to a single key on the keyboard. Because our template holds only digraphs, in the case of key hold latencies, each cell records the key hold latency of the first letter in the digraph (*i.e.*, cell "th" has key hold latencies of "t" only when the next letter typed is h).

To discard extreme latency values, we used a simple distance based outlier detection method where a latency value in a cell is considered an outlier if it does not have a predefined number of latencies (neighbors) within the neighborhood distance r. We calculated the predefined number of neighbors as α percent of the total number of latencies in the cell. In our experiment, we used $r = 100$ and $\alpha = 68\%$ for key interval and key press and $r = 80$ and $\alpha = 50\%$ for key hold. We used three verifiers: 1) relative ("R") verifier, 2) absolute ("A") verifier, and 3) similarity ("S") verifier. The "R" and "A" verifiers were proposed in [37] and the "S" verifier was proposed in [48].

Matching Pairs: In continuous verification, the user is free to type any text he/she desires. Therefore, some digraphs in a verification text may not have corresponding signatures in the template (*i.e.*, some cells in the template may be empty). The verifier outputs a score only after the verification text and the template have M number of digraphs in common. We refer to these M digraphs as *matching pairs*. We experimented with different M values.

CHAPTER 3

RELATED WORK AND MOTIVATION

3.1 Related Work and Motivation Behind the Symmetric Rejection Method

In Section 2.3, we discussed the importance of controlling the genuine reject rate in multi-stage biometric verification. However, to our knowledge, no work has focused on controlling the genuine reject rate in multi-stage biometric verification. In Table 3.1, we present a summary of the existing multi-stage biometric verification schemes. The first column of the Table 3.1 shows the studies focusing on multi-stage biometric verification. The second column specifies how each study selects the reject region. The third column specifies the domain of the thresholds. The fourth column specifies whether the verification score in i^{th}-stage is fused with the verification scores in previous stages.

Rejection methods proposed in [23], [24], and [25] select the *whole* confusion region as the reject region, which can cause a large proportion of genuine scores to be rejected because all scores in the confusion region are rejected indiscriminately, without considering to which part of the confusion region the score belongs. As a result, a genuine score falling in the portion of the confusion region where genuine scores outnumber impostor scores is treated the same way as a genuine score falling in the portion where impostor scores outnumber genuine.

Table 3.1: A summary of the existing multi-stage biometric verification schemes.

Studies	**How to select the reject region?**	**Threshold domain**	**Score in i^{th} stage fused with scores in previous stages?**
Marcialis et al. [23, 24] Zahir et al. [25]	The whole confusion region, where genuine and impostor scores overlap, is selected as the reject region.	Verification scores	No
Sansone et al. [22]	An effectiveness function is defined based on application, classification quality, error rates, and reject rates. A single threshold, which maximizes the effectiveness function, is selected by an exhaustive search. The selected threshold maximizes error-reject trade-off.	Classification reliability	No
Allano et al. [26] Takahashi et al. [27]	For given impostor pass rate and false alarm rate, two reject thresholds L and R are set using Wald's sequential probability ratio test [49]. The selected thresholds minimize the average number of stages required to verify.	Likelihood ratio between genuine score density and impostor score density.	Yes
Murakami et al. [28]	For a given impostor pass rate, a single threshold is set using the minimum log-likelihood ratio. The selected threshold minimizes the average number of stages required to verify.	Likelihood ratio between genuine score density and impostor score density.	Yes

Rejection method in [22] maximizes an effectiveness function defined on several parameters like classification quality, error rates, reject rates, etc.; however, it does not control the genuine reject rate.

Studies such as [26] and [27] select the reject region using the sequential probability ratio test (SPRT) and [28] selects the reject region using the minimum log-likelihood ratio (MLR). The reject regions based on SPRT and MLR do not indiscriminately reject scores in the confusion region. However, they require an accurate estimation of genuine and impostor score distributions, which is non-trivial because (1) it involves collecting biometric samples from a large number of users, which is an expensive task (see [50] and [51]) and (2) biometric scores can be dependent, which makes it only harder to estimate the distributions (see [9] and [52]).

In this dissertation, we develop a new rejection method, which *combines* the practical advantages of [23], [24], and [25] by directly estimating the reject region from scores, with the advantages of [26], [27], and [28], where genuine scores inside the reject region are not rejected indiscriminately.

3.2 Related Work and Motivation Behind the Proposed Fusion Framework

Two widely used modes of fusion are: (1) parallel mode and (2) serial mode. Parallel fusion approaches proposed in the literature can be broadly categorized into three groups: (1) arithmetic combination approach, (2) classification approach, and (3) density based approach. In the arithmetic combination approach, the individual matching scores are combined by performing some arithmetic operation such as summation, average, product, minimum, maximum, or median to generate a single fused score, which is used to make the verification decision [1], [18]. In the classification approach, a feature vector is built from the individual matching scores, which is then classified as genuine or impostor using some two-class classifier such as linear discriminant analysis, k-nearest neighbors, artificial neural network, or support vector machine [6], [19], [20]. In density based approach, a multi-dimensional density function is estimated from the matching scores and the verification decision is given based on the likelihood ratio test [4], [9], [21].

Serial fusion based multi-biometric verification systems proposed in the literature can be broadly categorized into two types: (1) serial *non score-fusion* based system (*e.g.*, [22], [23], [24], and [25]), in which the matching score obtained from the i^{th} biometric trait is not fused with the matching scores obtained from the previous biometric traits in the processing chain, and (2) serial *score-fusion* based system (*e.g.*, [26], [27], and [28]), in which the matching score obtained from the i^{th} biometric trait is fused with the matching scores obtained from the previous $i - 1$ biometric traits.

Parallel fusion based multi-biometric verification systems have received more attention from researchers because of their higher accuracy [2]. However, several recent studies such as [23], [24], [26], and [27] have questioned the applicability of parallel fusion based multi-biometric verification systems in many real world applications because parallel fusion can cause serious inconvenience to genuine users. For example, some genuine users could be accepted by the verification system by using only one biometric trait; however, in parallel fusion, he/she is bound to submit all of the n biometric traits (in the case of an n-biometric verification system). In such cases, parallel fusion turns out to be very time consuming and irritating. The problem becomes severe in the applications that involve a large population of users or a huge number of biometric transactions.

Serial fusion can provide convenience to the genuine users by allowing them to submit a subset of the biometric traits. For example, when a serial fusion based multi-biometric verification system gets enough evidence for classification after processing the first biometric trait, the user does not need to submit other biometric traits. A serial fusion based system can also allow the user to decide which biometric trait he/she would submit first [2]. Unfortunately, studies such as [26] and [36] show that serial fusion cannot reach the accuracy level of parallel fusion. Therefore, the applicability of the serial fusion is questionable in highly secured systems which require robustness to forgeries, robustness to enrollment problems, etc.

In this dissertation, we propose a new fusion framework for multi-biometric verification systems, which (1) achieves an accuracy level higher than parallel fusion, and at the same time (2) provides a significant amount of convenience to genuine users. In other words, we propose a fusion framework which combines the advantages of parallel fusion and serial fusion. Hence, our proposed fusion framework is applicable to high security applications as well as applications that involve a large population of users or a great number of biometric transactions.

3.3 Related Work and Motivation Behind the Proposed Framework of Continuous Keystroke Verification with Weak Templates

A number of transitory factors such as the state of mind (*e.g.*, anxiety), cognitive state (*e.g.*, inferring) while typing, physical conditions (*e.g.*, minor finger or hand injuries), and operational environment (*e.g.*, size of the keyboard and its position relative to the body) can affect typing behavior. To average-out the effect of such transitory factors, in keystroke verification, a user's template is built from keystroke samples collected in multiple enrollment sessions, each session occurring at a different point in time (i.e., sessions are spread over days, weeks, or even months).

Several studies in keystroke verification ([37], [38], and [39]) show that increasing the number of enrollment samples with "multi-session enrollment" lowers the verification error rates. In [37], each user provided 15 free text samples for enrollment in sessions occurring in different days, weeks, or months (no two enrollment samples were collected on the same day). Empirical results (in tables VIII and IX in [37]) show that the error rates for continuous keystroke verification significantly drop as more number of samples are used for enrollment. Study in [38] systematically evaluated three fixed text password based verifiers on data collected from 51 users. Each user typed 400 samples in 8 sessions, 50 samples per session, each session occurring in different days. Results (in Table 1 in [38]) show that all three verifiers achieve lower error rates with 100 and 200 enrollment samples than with 5 and 50 enrollment samples. (Note: 5 and 50 samples were collected in a single enrollment session while 100 and 200 samples were collected in multiple sessions.) Other keystroke verification studies which collected enrollment samples in multiple sessions are [39], [53], and [54].

A practical limitation with "multi-session enrollment" is that it assumes users are either available or solicitable multiple times. Though this assumption is usually true in a controlled laboratory setting, its validity is questionable in realistic cyber environments. For example, in cloud computing environments, thousands of users access system resources remotely (over the network) and it is unrealistic to expect that every user is available in a timely fashion to submit enrollment samples. Some users may even access the system so infrequently that repetitive collection of enrollment samples is impractical. With such users, a keystroke verification system is left to use a "weak" template, which is a template built from insufficient enrollment samples (*e.g.*, enrollment samples collected in a single session).

In this dissertation, we address the above challenge by introducing a new framework which consists of impostor score based normalization, impostor score based rejection, and fusion. Our goal is to minimize the equal error rates (EERs) of continuous keystroke verification when "weak" templates are used. We introduce a new formulation to incorporate the reject option in verification with weak templates and develop a new impostor score based rejection method called Order Statistic (OS) rejection method.

CHAPTER 4

SYMMETRIC REJECTION METHOD

In this chapter, we introduce a new rejection method called symmetric rejection method for multi-stage biometric verification. Compared to the existing rejection methods, the symmetric rejection method has two advantages: (1) it enables the system administrator to control genuine reject rate and (2) it allows the administrator to calculate the reject region directly from scores, without the need to estimate underlying probability density function. We analytically show how the symmetric rejection method reduces the false alarm rate and the impostor pass rate when it is used in a multi-stage biometric verification system. We validate our theory by experimenting on a four-stage biometric verification system. We compare the performance of the symmetric rejection method with two existing rejection methods: (1) SPRT-based method [26], which uses score-fusion and (2) Marcialis *et al.*'s method [23], which does not use score-fusion.

The rest of the chapter is organized as follows. In Section 4.1, we present the symmetric rejection method and describe how this method allows us to control the genuine reject rate and estimate the reject region directly from scores. In Section 4.2, we analyze the performance of the symmetric rejection method on a multi-stage verification system. In Section 4.3, we empirically validate our findings and claims.

In Section 4.4, we empirically compare the performance of the symmetric rejection method with SPRT-based [26] and Marcialis *et al.*'s [23] rejection methods.

4.1 Proposed Method

Let X be the set of genuine scores and Y be the set of impostor scores. Without loss of generality, we assume that the verifiers output real-valued dissimilar scores. If we classify the scores as genuine or impostor using the *traditional* verification rule (which uses a single threshold to give a binary "genuine/impostor" decision) and plot a DET-curve, we can find a threshold where the impostor pass rate and false alarm rate are equal. This error rate is called the equal error rate (EER) and we refer to the threshold where EER occurs as *EER-threshold*. Because in the EER-threshold, the impostor pass rate and false alarm rate are equal, it is expected that the scores that surround the EER-threshold are most confusing and therefore are most likely to get a wrong verification decision. Considering this, in the symmetric rejection method, we take the EER-threshold as the center of rejection and reject the scores that surround it.

We demonstrate an EER-threshold in Figure 4.1, where ZE_2 is the genuine score region, E_1O is the impostor score region, E_1E_2 is the confusion region, and B is the EER-threshold. Note that the impostor scores in the left side of the EER-threshold (impostor scores in E_1B in Figure 4.1) give the impostor pass rate and the genuine scores in the right side of the EER-threshold (genuine scores in BE_2 in Figure 4.1) give the false alarm rate. In the symmetric rejection, our goal is to reduce both impostor pass rate and false alarm rate, and at the same time maintain a specified genuine reject rate. Considering this, we reject scores such that the proportion of impostor scores rejected from the left side of the EER-threshold is equal to the proportion of the genuine scores rejected from the right side of the EER-threshold. Below, we explain the symmetric rejection method with the help of Figure 4.1.

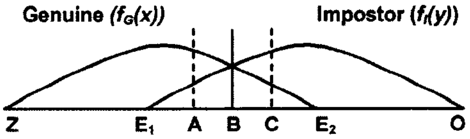


Figure 4.1: Illustration of the symmetric rejection. AC is the symmetric reject region, ZE_2 is the genuine score region, E_1O is the impostor score region, E_1E_2 is the confusion region, and B is the EER-threshold.

In Figure 4.1, $f_G(x)$ and $f_I(y)$ are the genuine and impostor score distributions, $[Z, O]$ is the scoreline, E_1E_2 is the confusion region, and B is the EER-threshold. We select the reject region AC such that B is the center of rejection and the proportion of impostor scores in AB is equal to the proportion of genuine scores in BC, where $A \in [E_1, B)$ and $C \in (B, E_2]$. We call AC the symmetric reject region. The proportion of impostor scores in AB is calculated by (number of impostor scores in AB)/(total number of impostor scores) and the proportion of genuine scores in BC is calculated by (number of genuine scores in BC)/(total number of genuine scores).

Assumption Made in Our Formulation and Related Proofs: In our formulation of the symmetric rejection method and its related proofs that follow, we assume 1) $f_G(x)$ is monotonically decreasing inside the confusion region and $f_I(y)$ is monotonically increasing inside the confusion region, and 2) $f_G(x)$ and $f_I(y)$ are continuous throughout the scoreline $[Z, O]$. For example, in Figure 4.2, $f_G(x)$ and

$f_I(y)$ in (a), (b), (c), and (d) follow the assumption. However, $f_G(x)$ and $f_I(y)$ in (e) do not follow the assumption because there are ups and downs within the confusion region E_1E_2.

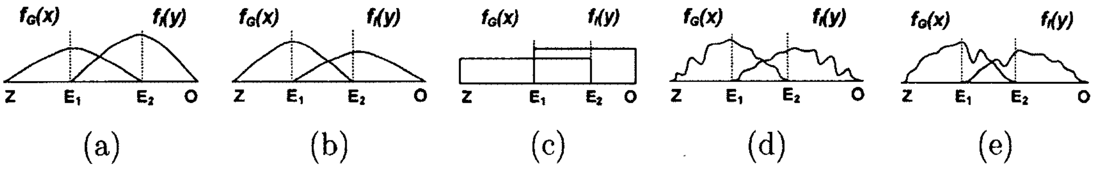
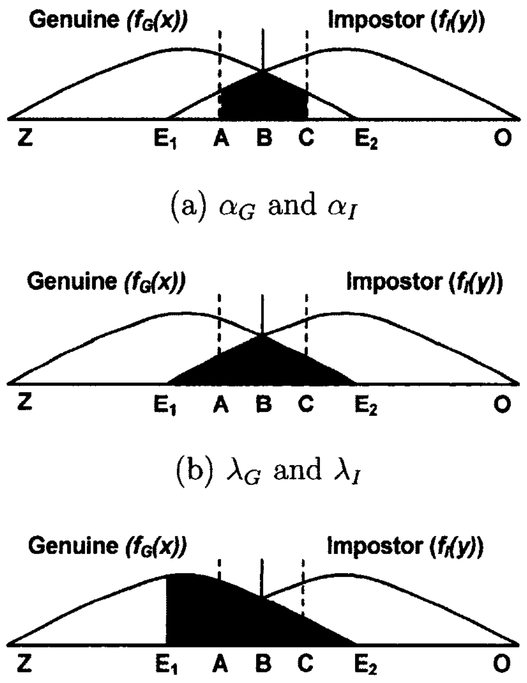


Figure 4.2: Examples of $\{f_G(x), f_I(y)\}$ following/not following the assumption that $f_G(x)$ is monotonically decreasing and $f_I(y)$ is monotonically increasing inside the confusion region E_1E_2. $f_G(x)$ and $f_I(y)$ in (a), (b), (c), and (d) follow the assumption. However, $f_G(x)$ and $f_I(y)$ in (e) do not follow the assumption because there are ups and downs in the confusion region E_1E_2.

While the above assumption simplifies our proofs, we note that our assumptions are true for a wide range of distributions, including Gaussian, certain parameters of beta, binomial, and beta-binomial, and Gaussian mixture model (except when a mode is inside the confusion region). The above mentioned distributions have also been used in various studies to model score distributions of various biometric modalities (see [9], [48], [55], [56], and [57]).

Notation and Symbols: Below, we introduce some notations and symbols corresponding to the symmetric rejection. We use Figure 4.3 to illustrate the notation.

- α_I: The proportion of impostor scores in AB (Figure 4.3a).
- α_G: The proportion of genuine scores in BC (Figure 4.3a).
- λ_I: The proportion of impostor scores in E_1B (Figure 4.3b).
- λ_G: The proportion of genuine scores in BE_2 (Figure 4.3b).
- K: The proportion of genuine scores in the confusion region E_1E_2 (Figure 4.3c).



(a) α_G and α_I

(b) λ_G and λ_I

(c) K

Figure 4.3: Illustration of α_G, α_I, λ_G, λ_I, and K.

4.1.1 Estimating Reject Region

The symmetric rejection method chooses the reject region such that $\alpha_G = \alpha_I$. By setting α_G at different values, we can get different symmetric reject regions. The minimum possible value of α_G is zero (when no score is rejected) and the maximum possible value of α_G is λ_G (when all scores in the confusion region are rejected). Based on the idea of symmetric rejection, we devised an algorithm, "Algorithm 1", which calculates the symmetric reject region AC from a given α_G. The benefit of Algorithm 1 is that it calculates the reject region ***directly from scores***, *i.e.*, it does not need to estimate the underlying probability density function of the scores. Below, we briefly explain Algorithm 1.

Algorithm 1 : Estimating Symmetric Reject Region.

Input: $G[1:M]$: array of M genuine scores,
$I[1:N]$: array of N impostor scores, and
α_G: proportion of genuine scores in BC (see Figure 4.1).
Output: Symmetric reject region AC

```
$G_{sorted}[1:M] \leftarrow$ sorted $G[1:M]$; /*Sort in ascending order*/
$I_{sorted}[1:N] \leftarrow$ sorted $I[1:N]$; /*Sort in ascending order*/
$B \leftarrow funcEERThreshold(G_{sorted}, I_{sorted})$; /*Calculate the EER before exercising reject option and return the EER-threshold.*/
$iBG_{sorted} \leftarrow funcFindIndexOfB(G_{sorted}, B)$; /*Search $B$ in $G_{sorted}$ and return the corresponding index. If $B$ is not found, return the index of the closest score in $G_{sorted}$ that is greater than $B$.*/
$iBI_{sorted} \leftarrow funcFindIndexOfB(I_{sorted}, B)$; /*Search $B$ in $I_{sorted}$ and return the corresponding index. If $B$ is not found, return the index of the closest score in $I_{sorted}$ that is less than $B$.*/
$nImpAB \leftarrow N * \alpha_G$; /*Calculate the number of impostor scores in $AB$.*/
$nGenBC \leftarrow M * \alpha_G$; /*Calculate the number of genuine scores in $BC$.*/
$iA \leftarrow iBI_{sorted} - nImpAB{+}1$; /*Calculate the index of $A$ in $I_{sorted}$.*/
$iC \leftarrow iBG_{sorted} + nGenBC\text{-}1$; /*Calculate the index of $C$ in $G_{sorted}$.*/
$A \leftarrow I_{sorted}[iA]$; /*Get the value of $A$.*/
$C \leftarrow G_{sorted}[iC]$; /*Get the value of $C$.*/
return $AC$;
```

In steps 1 and 2, we sort the scores in ascending order. Arrays G_{sorted} and I_{sorted} store the sorted genuine and impostor scores, respectively. In line 3, we use the function *funcEERThreshold*() to calculate the EER-threshold. In line 4, we find the location (index) of B in the array G_{sorted}. In line 5, we find the location of B in I_{sorted}. In line 6, we calculate the number of impostors to be rejected from the left side of B, which we store in variable $nImpAB$. Note that to calculate the value of $nImpAB$, we multiplied N with α_G instead of α_I. This is valid because in symmetric rejection, $\alpha_I = \alpha_G$. In line 7, we calculate the number of genuine scores to be rejected from the right side of B. In lines 8 and 9, we find the locations of A and C in arrays I_{sorted} and G_{sorted}, respectively. In lines 10 and 11, we extract the value of A and C from the corresponding arrays.

In the following section, we discuss how we determine the value of α_G.

4.1.2 Determination of α_G

We determine the value of α_G based on our desired genuine reject rate. To this end, we have derived a relationship between α_G and the upper bound for the genuine reject rate. The relationship is as follows: *when* α_G *is equal to* $\rho\lambda_G$, *where* ρ *is a rational number such that* $0 < \rho \leq 1$, *the upper bound for the genuine reject rate is* ρK. The derivation of this relationship is given in Section 4.1.3.

Below, we show how the above relationship helps us to determine α_G for a desired upper bound for the genuine reject rate.

Let x denote the upper bound for the genuine reject rate. Then we can rewrite the above relationship as follows: $x = \rho K$ when $\alpha_G = \rho\lambda_G$. Or alternatively,

$$x = \frac{\alpha_G}{\lambda_G}K = \frac{K}{\lambda_G}\alpha_G. \tag{4.1}$$

Because K and λ_G are constant for a given genuine and impostor score set, x is a strictly monotonic function of α_G. Let $f : \alpha_G \rightarrow x$ represent the function. We define f as follows:

$$x = f(\alpha_G) = \frac{K}{\lambda_G}\alpha_G. \tag{4.2}$$

Because x is a strictly monotonic function of α_G, we can calculate the inverse of f. Below, we show the calculation of $f^{-1} : x \rightarrow \alpha_G$.

$$\alpha_G = \rho\lambda_G = \frac{x}{K}\lambda_G = \frac{\lambda_G}{K}x.$$

Or alternatively,

$$\alpha_G = f^{-1}(x) = \frac{\lambda_G}{K}x. \tag{4.3}$$

We use (4.3) to determine α_G for a desired upper bound for the genuine reject rate, x.

Step-by-step procedure to determine α_G***:*** Below, we give a *three-step procedure* to determine α_G for a desired upper bound for the genuine reject rate, x.

- *Step 1:* Find the confusion region E_1E_2 and calculate K by (number of genuine scores in E_1E_2)/(total number of genuine scores).
- *Step 2:* Find the EER-threshold B and calculate λ_G by (number of genuine scores in BE_2)/(total number of genuine scores).
- *Step 3:* Calculate $\alpha_G = \frac{\lambda_G}{K}x$.

For example, let $K = 0.5$ (*i.e.*, 50% of genuine scores lie inside the confusion region E_1E_2) and $\lambda_G = 0.2$ (*i.e.*, 20% genuine scores lie inside BE_2). Let us assume that we want to find a reject region which will not reject more than 10% of the genuine scores, *i.e.*, $x = 0.1$. Then using the above procedure, $\alpha_G = (0.2/0.5)*0.1 = 0.04$.

Summary of Section 4.1.1 and Section 4.1.2: In Section 4.1.1, we have shown how to estimate the reject region AC from a given α_G. In Section 4.1.2, we have shown how to determine α_G from a specified upper bound for the genuine reject rate. If we combine Section 4.1.1 and Section 4.1.2, we find a way to estimate the reject region AC from a specified upper bound for the genuine reject rate x, which involves the following two steps:

- *STEP I:* Determine α_G for the specified upper bound x using the *three-step procedure* described above.
- *STEP II:* Use the value of α_G estimated in STEP I to calculate the reject region AC using Algorithm 1.

In the following section, we give the derivation of the relationship between α_G and the upper bound for the genuine reject rate.

4.1.3 Derivation of Relationship between α_G and the Upper Bound for Genuine Reject Rate

In this section, we derive the following relationship between α_G and upper bound for the genuine reject rate: *when α_G is equal to $\rho\lambda_G$, where ρ is a rational number such that $0 < \rho \leq 1$, the upper bound for the genuine reject rate is ρK.*

For simplicity, here, we will show the derivation for a specific value of α_G. A generalized derivation is given in Appendix A.

Let $\alpha_G = 0.75\lambda_G$, *i.e.*, $\rho = 0.75$. We will present ρ by fractions $\frac{q}{m}$ such that q and m are positive integers. Here, we present ρ by $\frac{3}{4}$, *i.e.*, $q = 3$ and $m = 4$.

We use Figure 4.4 to explain the derivation. In Figure 4.4, let B be the EER-threshold and E_1E_2 be the confusion region. λ_G is the proportion of the genuine scores in BE_2 and λ_I is the proportion of the impostor scores in E_1B. Because $m = 4$, we divide BE_2 into 4 parts such that the proportion of the genuine scores in each part is $\frac{\lambda_G}{4}$. Similarly, we divide E_1B into 4 parts such that the proportion of the impostor scores in each part is $\frac{\lambda_I}{4}$. Let b_{G_1}, b_{G_2}, b_{G_3}, and b_{G_4} be the proportions of the genuine scores in the four parts of E_1B (see Figure 4.4). Because $f_G(x)$ is monotonically decreasing and $f_I(y)$ is monotonically increasing inside the confusion region, b_{G_1}, b_{G_2}, b_{G_3}, and b_{G_4} are related as: $b_{G_1} \leq b_{G_2} \leq b_{G_3} \leq b_{G_4}$.

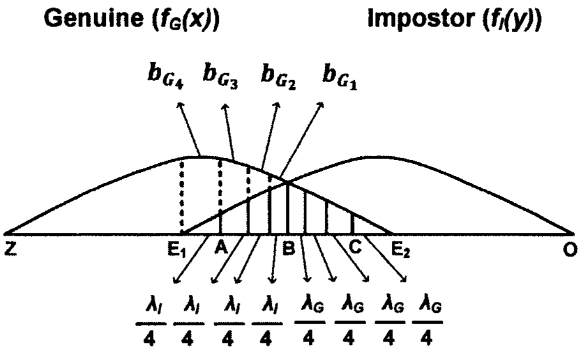

Genuine ($f_G(x)$)
Impostor ($f_I(y)$)
b_{G_4}
b_{G_3}
b_{G_2}
b_{G_1}
Z
E₁
A
B
C
E₂
O
$\frac{\lambda_I}{4}$ $\frac{\lambda_I}{4}$ $\frac{\lambda_I}{4}$ $\frac{\lambda_I}{4}$ $\frac{\lambda_G}{4}$ $\frac{\lambda_G}{4}$ $\frac{\lambda_G}{4}$ $\frac{\lambda_G}{4}$

Figure 4.4: Dividing E_1B and BE_2 into four parts. E_1B is divided into 4 parts such that the proportion of impostor scores in each part is $\frac{\lambda_I}{4}$. Similarly, BE_2 is divided into 4 parts such that the proportion of genuine scores in each part is $\frac{\lambda_G}{4}$. b_{G_1}, b_{G_2}, b_{G_3}, and b_{G_4} are the proportions of genuine scores in the four parts of E_1B.

Using the notation in Figure 4.4, we can present the proportion of genuine scores in the confusion region E_1E_2, K, as follows:

$$K = b_{G_1} + b_{G_2} + b_{G_3} + b_{G_4} + \lambda_G. \tag{4.4}$$

We can calculate the genuine reject rate (GRR) as follows:

$$GRR = \text{Proportion of genuine scores in } AC, \tag{4.5}$$

where AC is the symmetric reject region. Because B is a threshold in between A and C (see Figure 4.4), we can rewrite (4.5) as follows:

$$GRR = \text{Proportion of genuine scores in } AB + \text{Proportion of genuine scores in } BC. \tag{4.6}$$

Now we will explain what happens when $\alpha_G = \frac{3}{4}\lambda_G = 3\frac{\lambda_G}{4}$. Because α_G (*i.e.*, proportion of genuine scores in BC) is equal to 3 times $\frac{\lambda_G}{4}$, following the symmetric rejection rule, α_I (*i.e.*, proportion of impostor scores in AB) is equal to 3 times $\frac{\lambda_I}{4}$. As a result, the proportion of the genuine scores in AB is equal to $b_{G_1} + b_{G_2} + b_{G_3}$ (see Figure 4.4). Hence, we can rewrite (4.6) as follows:

$$\begin{aligned} GRR &= b_{G_1} + b_{G_2} + b_{G_3} + 3\frac{\lambda_G}{4} \\ &= \frac{3}{4}\{\frac{4}{3}(b_{G_1} + b_{G_2} + b_{G_3}) + \lambda_G\} \\ &= 0.75\{\frac{4}{3}(b_{G_1} + b_{G_2} + b_{G_3}) + \lambda_G\}. \end{aligned} \tag{4.7}$$

Now we will show that $\frac{4}{3}(b_{G_1} + b_{G_2} + b_{G_3}) \leq b_{G_1} + b_{G_2} + b_{G_3} + b_{G_4}$. For contradiction, we assume that $\frac{4}{3}(b_{G_1} + b_{G_2} + b_{G_3}) > b_{G_1} + b_{G_2} + b_{G_3} + b_{G_4}$. This implies that

$$4(b_{G_1} + b_{G_2} + b_{G_3}) > 3(b_{G_1} + b_{G_2} + b_{G_3} + b_{G_4}).$$

After algebraic manipulation, we get

$$b_{G_1} + b_{G_2} + b_{G_3} > 3b_{G_4}.$$

However, this is impossible because $b_{G_1} \leq b_{G_2} \leq b_{G_3} \leq b_{G_4}$. Therefore, the statement $\frac{4}{3}(b_{G_1} + b_{G_2} + b_{G_3}) \leq b_{G_1} + b_{G_2} + b_{G_3} + b_{G_4}$ is true. Hence, we can rewrite (4.7) as follows:

$$GRR \leq 0.75(b_{G_1} + b_{G_2} + b_{G_3} + b_{G_4} + \lambda_G). \tag{4.8}$$

Or alternatively, $GRR \leq 0.75K$ because $K = b_{G_1} + b_{G_2} + b_{G_3} + b_{G_4} + \lambda_G$. That is, the upper bound for the genuine reject rate is $0.75K$.

4.2 Performance of Symmetric Rejection Method on a Multi-stage Verification System

In this section, we analytically show how the symmetric rejection method reduces the false alarm rate and the impostor pass rate when it is used in a multi-stage biometric verification system. For simplicity, here, we perform the analysis on a three-stage verification system (a similar analysis is applicable to an n-stage verification system).

Let v_1, v_2, and v_3 be three verifiers such that v_1 performs better than v_2 and v_2 performs better than v_3 in terms of equal error rate, *i.e.*, $EER_1 < EER_2 < EER_3$, where EER_1, EER_2, and EER_3 are the equal error rates of v_1, v_2, and v_3, respectively. We model a three-stage verification system such that v_1 is placed in the first stage, v_2 in the second stage, and v_3 in the third (final) stage. The first and second stages use the symmetric rejection method and the third stage uses the threshold where EER_3 occurs, to give verification decisions.

We expect the performance of the three-stage verification system (described above) will be better than the top performing individual verifier v_1. That is, the false alarm rate and the impostor pass rate obtained by the three-stage verification system will be less than EER_1. In this section, we show how the symmetric rejection method ensures this.

Let FAR_3 denote the false alarm rate and IPR_3 denote the impostor pass rate of the three-stage verification system described above. We will prove that

$$FAR_3 \text{ and } IPR_3 \text{ will be less than } EER_1 \text{ if } \frac{EER_3}{EER_1} < \frac{1}{K_1}, \tag{4.9}$$

where K_1 is the proportion of the genuine scores within the confusion region of the top performing verifier v_1.

For example, let $K_1 = 0.25$. That is, 25% of the genuine scores originated by verifier v_1 lie inside the confusion region. In this case, FAR_3 and IPR_3 will be less than EER_1 if (EER_3/EER_1) is less than (1/0.25) or 4.

We give the proof of statement in (4.9) in Section 4.2.1. The proof uses the following lemma, which states that the symmetric rejection method reduces both the false alarm rate and the impostor pass rate when it is used in an *individual* verifier.

Lemma 4.1. *Let v_i be a verifier with equal error rate EER_i. Let FAR_i^r and IPR_i^r be the false alarm rate and impostor pass rate, respectively, obtained by applying the symmetric rejection method on the verification scores produced by v_i. Then, both FAR_i^r and IPR_i^r are less than EER_i.*

The proof for Lemma 4.1 is given in Appendix B.

4.2.1 Proof that FAR_3 and IPR_3 will be less than EER_1 if $\frac{EER_3}{EER_1} < \frac{1}{K_1}$

Below, we prove that FAR_3 will be less than EER_1 if $(EER_3/EER_1) < \frac{1}{K_1}$. The proof for IPR_3 is similar to the proof for FAR_3.

We use Figure 4.5 to demonstrate FAR_3. Let n_G denote the total number of genuine scores submitted to verifier v_1 in Stage 1. Verifier v_1 uses the symmetric rejection method. Let v_1 correctly declare $n_{GG,1}$ genuine scores as genuine, erroneously declare $n_{GI,1}$ genuine scores as impostor, and reject $n_{GR,1}$ genuine scores. Hence, the number of genuine scores submitted to verifier v_2 in Stage 2 is equal to $n_{GR,1}$. Verifier v_2 uses the symmetric rejection method, correctly declares $n_{GG,2}$ genuine scores as genuine, erroneously declares $n_{GI,2}$ genuine scores as impostor, and rejects $n_{GR,2}$ genuine scores. Hence, the number of genuine scores submitted to verifier v_3 in Stage 3 is equal to $n_{GR,2}$. Verifier v_3 gives the binary decision using the threshold where EER_3 occurs, correctly declares $n_{GG,3}$ genuine scores as genuine, and erroneously declares $n_{GI,3}$ genuine scores as impostor.

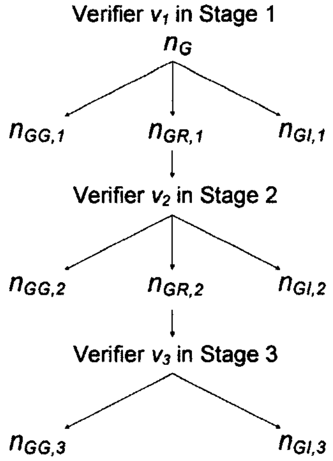


Figure 4.5: A three-stage verification system to demonstrate FAR_3. Because we demonstrate FAR_3, we only show the flow of genuine scores.

From Figure 4.5, the number of genuine scores erroneously declared as impostor by the three-stage system is equal to $n_{GI,1} + n_{GI,2} + n_{GI,3}$. Therefore, the false alarm rate of the three-stage system is

$$FAR_3 = \frac{n_{GI,1} + n_{GI,2} + n_{GI,3}}{n_G}. \tag{4.10}$$

We use Figure 4.1 to demonstrate EER_1. Let (1) the genuine scores and impostor scores in Figure 4.1 originate from verifier v_1, (2) the number of genuine scores in ZE_2 is equal to n_G, and (3) B be the threshold where EER_1 occurs. Then we can present EER_1 as follows:

$$\begin{aligned} EER_1 &= \text{Proportion of genuine scores in } BE_2 \\ &= \frac{\#\text{ of genuine scores in } BE_2}{n_G}. \end{aligned} \tag{4.11}$$

In Figure 4.1, let AC be the symmetric reject region used by Stage 1. Then, the genuine scores that lie inside CE_2 are erroneously declared as an impostor by Stage 1. That is,

$$\#\text{ of genuine scores in } CE_2 = n_{GI,1}.$$

Hence, we can present the numerator on the right side of (4.11) as:

$$\begin{aligned} \#\text{ of genuine scores in } BE_2 &= \#\text{ of genuine scores in } BC + \#\text{ of genuine scores} \\ &\quad \text{in } CE_2 \\ &= \#\text{ of genuine scores in } BC + n_{GI,1}. \end{aligned}$$

Therefore, we can rewrite (4.11) as:

$$EER_1 = \frac{n_{GI,1} + \#\text{ of genuine scores in } BC}{n_G}. \tag{4.12}$$

By comparing (4.10) and (4.12), we find that FAR_3 will be less than EER_1 if the following statement is true:

$$n_{GI,2} + n_{GI,3} < \#\text{ of genuine scores in } BC$$

or alternatively,

$$\frac{n_{GI,2} + n_{GI,3}}{n_{GG,2} + n_{GI,2} + n_{GG,3} + n_{GI,3}} < \frac{\#\text{ of genuine scores in } BC}{\#\text{ of genuine scores in } AC} \quad (4.13)$$

because $n_{GG,2} + n_{GI,2} + n_{GG,3} + n_{GI,3} = n_{GR,1} = \#$ of genuine scores in AC.

Below, we discuss how the statement in (4.13) is true.

First, we analyze the left side of (4.13). In (4.13), $n_{GG,2} + n_{GI,2} + n_{GG,3} + n_{GI,3}$ is equal to the number of genuine scores submitted to verifier v_2 in Stage 2 (see Figure 4.5). Stage 2 rejects $n_{GG,3} + n_{GI,3}$ genuine scores, correctly declares $n_{GG,2}$ genuine scores as genuine, and erroneously declares $n_{GI,2}$ genuine scores as impostor by using the symmetric rejection method. Therefore, $n_{GI,2}/(n_{GG,2} + n_{GI,2})$ in (4.13) represents the false alarm rate of verifier v_2 obtained by the symmetric rejection method (FAR_2^r). That is,

$$\frac{n_{GI,2}}{n_{GG,2} + n_{GI,2}} = FAR_2^r.$$

In (4.13), $n_{GG,3} + n_{GI,3}$ is equal to the number of genuine scores submitted to verifier v_3 in Stage 3 (see Figure 4.5). Stage 3 correctly declares $n_{GG,3}$ genuine scores as genuine, and erroneously declares $n_{GI,3}$ genuine scores as impostor by using the threshold where EER_3 occurs. Therefore, $n_{GI,3}/(n_{GG,3} + n_{GI,3})$ in (4.13) represents EER_3. That is,

$$\frac{n_{GI,3}}{n_{GG,3} + n_{GI,3}} = EER_3.$$

Because (1) $FAR_2^r < EER_2$ (according to Lemma 4.1) and (2) $EER_2 < EER_3$ (according to our design of the three-stage verification system), we deduce that $FAR_2^r < EER_3$. That is,

$$\frac{n_{GI,2}}{n_{GG,2} + n_{GI,2}} < \frac{n_{GI,3}}{n_{GG,3} + n_{GI,3}}. \tag{4.14}$$

It is easy to show that if $\frac{a}{b} < \frac{c}{d}$, then $\frac{a+c}{b+d} < \frac{c}{d}$. By using this property in (4.14), we find that

$$\frac{n_{GI,2} + n_{GI,3}}{n_{GG,2} + n_{GI,2} + n_{GG,3} + n_{GI,3}} < \frac{n_{GI,3}}{n_{GG,3} + n_{GI,3}}$$

or alternatively,

$$\frac{n_{GI,2} + n_{GI,3}}{n_{GG,2} + n_{GI,2} + n_{GG,3} + n_{GI,3}} < EER_3 \tag{4.15}$$

because $EER_3 = n_{GI,3}/(n_{GG,3} + n_{GI,3})$.

Now we analyze the right side of (4.13). We can rewrite the right side of (4.13) as follows:

$$\frac{\#\text{ of gen. scores in } BC}{\#\text{ of gen. scores in } AC} = \frac{\text{Proportion of gen. scores in } BC}{\text{Proportion of gen. scores in } AC} = \frac{\alpha_{G_1}}{GRR_1}.$$

Here, we use subscript '1' to specify that these α_G and GRR correspond to verifier v_1 in Stage 1.

According to Section 4.1.3, α_{G_1} and GRR_1 are related as follows: when $\alpha_{G_1} = \rho\lambda_{G_1}$, $0 < \rho \leq 1$, the upper bound of GRR_1 is ρK_1, where λ_{G_1} is the proportion of genuine scores in BE_2 and K_1 is the proportion of genuine scores in the confusion region E_1E_2. Here, we use subscript '1' to specify that these λ_G and K correspond to verifier v_1 in Stage 1.

Because ρK_1 is the upper bound for GRR_1 (*i.e.*, maximum possible value of GRR_1), if we divide $\alpha_{G_1}(=\rho\lambda_{G_1})$ by ρK_1, we will find the minimum value of $\frac{\alpha_{G_1}}{GRR_1}$. Hence, the minimum value of $\frac{\alpha_{G_1}}{GRR_1}$ is equal to $\frac{\rho\lambda_{G_1}}{\rho K_1}$ or $\frac{\lambda_{G_1}}{K_1}$. In other words,

$$\frac{\alpha_{G_1}}{GRR_1} \geq \frac{\lambda_{G_1}}{K_1}. \tag{4.16}$$

Because λ_{G_1} = proportion of genuine scores in $BE_2 = EER_1$, we can rewrite (4.16) as follows:

$$\frac{\alpha_{G_1}}{GRR_1} \geq \frac{EER_1}{K_1}$$

or alternatively,

$$\frac{\#\text{ of gen. scores in } BC}{\#\text{ of gen. scores in } AC} \geq \frac{EER_1}{K_1}. \tag{4.17}$$

Now we are ready to state how the statement in (4.13) is true.

Because (1) the left-hand side of (4.13) is less than EER_3 (see (4.15)) and (2) the right-hand side of (4.13) is greater than or equal to $\frac{EER_1}{K_1}$ (see (4.17)), we deduce that the statement in (4.13) will be true if

$$EER_3 < \frac{EER_1}{K_1}$$

or alternatively,

$$\frac{EER_3}{EER_1} < \frac{1}{K_1}.$$

Therefore, we conclude that FAR_3 will be less than EER_1 if $\frac{EER_3}{EER_1} < \frac{1}{K_1}$.

4.2.2 General Case: An n-stage Verification System

Let v_1, v_2, $\cdots$, v_n be n verifiers such that $EER_1 < EER_2 < \cdots < EER_n$, where EER_i is the equal error rate of verifier v_i, for $i = 1, 2, \cdots, n$. We model an

n-stage verification system such that v_1 is placed in the first stage, v_2 is placed in the second stage, and so on, until v_n is placed in the n^{th} stage. Stages 1 to $n-1$ use the symmetric rejection method and the n^{th}-stage uses the threshold where EER_n occurs to give verification decisions. Let FAR_n represent the false alarm rate and IPR_n represent the impostor pass rate of the n-stage system. Then according to (4.9), FAR_n and IPR_n will be less than EER_1 if the following condition is satisfied:

$$\frac{EER_n}{EER_1} < \frac{1}{K_1}, \tag{4.18}$$

where K_1 is the proportion of genuine scores within the confusion region of the top performing verifier v_1.

4.3 Empirical Validation

We empirically validated: (1) when α_G is equal to $\rho\lambda_G$, where ρ is a rational number such that $0 < \rho \leq 1$, the upper bound for the genuine reject rate is ρK and (2) FAR_n and IPR_n will be less than EER_1 if $(EER_n/EER_1) < \frac{1}{K_1}$.

We did our experiments using NIST Biometric Scores Set Release 1 (BSSR 1) [41]. The NIST-BSSR1 database consists of three score sets–1) fingerprint-face, 2) fingerprint, and 3) face. We experimented with the score set fingerprint-face. The fingerprint-face set consists of face and fingerprint scores from the same set of 517 individuals. For each individual, the set contains one score from the comparison of two right index fingerprints, one score from the comparison of two left index fingerprints, and two scores (from two separate face verifiers, namely, C and G) from the comparison of two frontal faces. The scores of the right and left index fingerprints were generated using the same fingerprint verifier. The fingerprint images and the face images used to compute the scores were collected from the same person and at the same time.

We divided the fingerprint-face score set into four disjoint subsets–(1) LI: set of scores of left index fingerprints, (2) RI: set of scores of right index fingerprints, (3) C: set of scores from face verifier C, and (4) G: set of scores from face verifier G. We used the scores of the first 259 individuals from each of these sets (LI, RI, C, and G) for training and the scores from the rest of the 258 individuals in testing. The number of genuine scores and impostor scores in training and testing sets are presented in Table 4.1.

Table 4.1: The number of genuine and impostor scores in training and testing sets of LI, RI, C, and G.

	Score sets (LI/RI/C/G)	
	Training sets	Testing sets
Genuine scores	259*1	258*1
Impostor scores	259*516	258*516

4.3.1 Validation of Our Derived Relationship between α_G and the Upper Bound for Genuine Reject Rate

In Section 4.1.3, we theoretically proved that when α_G is equal to $\rho\lambda_G$, where ρ is a rational number such that $0 < \rho \leq 1$, the upper bound for the genuine reject rate is ρK. In this section, we empirically validate the statement.

Validation Procedure: For different values of α_G, we (1) estimate the theoretical upper bound for the genuine reject rate (ρK), (2) apply the symmetric rejection method on verification scores and find the genuine reject rate (GRR), and

(3) compare the genuine reject rate obtained from the experiment with its theoretical upper bound to see whether the experimental value of the genuine reject rate is less than its theoretical upper bound. Below, we give the details of the experiment.

We experimented with 10 different values of α_G on score set LI, RI, C, and G . We set the initial value of α_G to $0.1\lambda_G$. Then we incremented the value of α_G 9 times, each time by $0.1\lambda_G$, up to λ_G, which is the maximum possible value of α_G.

For each α_G, we estimated the theoretical upper bound for the genuine reject rate (ρK). Table 4.2 shows the values of K in score sets LI, RI, C, and G, estimated on training data. Below, we demonstrate estimation of theoretical upper bound (ρK) in the case of score sets LI, RI, C, and G.

Table 4.2: Ks and λ_Gs of score sets LI, RI, C, and G, estimated on training data.

	LI	RI	C	G
K	0.3707	0.1660	0.5792	0.4054
λ_G	0.0772	0.0547	0.0463	0.0579

Let α_G is equal to $0.1\lambda_G$. Then, in the case of score set LI, the theoretical upper bound (ρK) is 0.1*0.3707 or 0.0371. Similarly, in the case of RI, ρK is 0.1*0.1660 or 0.0166, in the case of C, ρK is 0.1*0.5792 or 0.0579, and in the case of G, ρK is 0.1*0.4054 or 0.0405.

We obtained the experimental values of the genuine reject rate (GRR) as follows: for each value of α_G, we estimated the symmetric reject region AC from the training data, using Algorithm 1. We applied the reject region AC on the testing data and calculated GRR by (number of genuine scores falling inside AC)/(total number of genuine scores). In Table 4.3, we compare the experimental values of the genuine reject rates (GRRs) with their theoretical upper bounds (ρKs).

Table 4.3: Comparing experimental values of genuine reject rate (GRRs) with their theoretical upper bounds (ρKs). We performed experiments on four score sets LI, RI, C, and G with 10 different values of α_G.

α_G	LI		RI		C		G	
	ρK	GRR	ρK	GRR	ρK	GRR	ρK	GRR
$0.1\lambda_G$	0.0371	0.0116	0.0166	0.0154	0.0579	0.0039	0.0405	0.0116
$0.2\lambda_G$	0.0741	0.0116	0.0332	0.0194	0.1158	0.0078	0.0811	0.0155
$0.3\lambda_G$	0.1112	0.031	0.0498	0.0194	0.1737	0.0155	0.1216	0.0349
$0.4\lambda_G$	0.1483	0.0349	0.0664	0.0349	0.2317	0.0233	0.1622	0.0581
$0.5\lambda_G$	0.1853	0.0426	0.083	0.0388	0.2896	0.0271	0.2027	0.062
$0.6\lambda_G$	0.2224	0.0736	0.0996	0.0465	0.3475	0.031	0.2432	0.0698
$0.7\lambda_G$	0.2595	0.0814	0.1162	0.062	0.4054	0.0388	0.2838	0.093
$0.8\lambda_G$	0.2965	0.0891	0.1328	0.0698	0.4633	0.0775	0.3243	0.1124
$0.9\lambda_G$	0.3336	0.1434	0.1494	0.0736	0.5212	0.1008	0.3649	0.1395
λ_G	0.3707	0.3527	0.1660	0.1660	0.5792	0.5581	0.4054	0.3721

Our observations from Table 4.3 are listed below:

- ***Observation 1:*** Experimental values of genuine reject rate, GRRs, are less than the theoretical upper bounds, ρKs, for all α_Gs, and this happens to all score sets LI, RI, C, and G. Hence, the results of our experiment validate our derived relationship: when α_G is equal to $\rho\lambda_G$, where ρ is a rational number such that $0 < \rho \leq 1$, the upper bound for the genuine reject rate is ρK.
- ***Observation 2:*** Experimental values of the genuine reject rate, GRRs, are close to the theoretical upper bounds, ρKs, for large α_Gs. For example, when $\alpha_G = \lambda_G$, GRR is equal to 95.14% of ρK in the case of score set LI, 100% of ρK in the case of score set RI, 96.54% of ρK in the case of score set C, and 91.79% of ρK in the case of score set G. However, the tightness of the upper

bound reduces when α_Gs are small. For example, when $\alpha_G = 0.1\lambda_G$, GRR is equal to 31.27% of ρK in the case of score set LI, 92.77% of ρK in the case of score set RI, 6.74% of ρK in the case of score set C, and 28.64% of ρK in the case of score set G. On average, GRR is equal to 35.40% of ρK in the case of score set LI, 59.13% of ρK in the case of score set RI, 19.27% of ρK in the case of score set C, and 36.90% of ρK in the case of score set G.

4.3.2 Validation of the Statement: FAR_n and IPR_n will be less than EER_1 if $\frac{EER_n}{EER_1} < \frac{1}{K_1}$

In Section 4.2, we theoretically proved that FAR_n and IPR_n will be less than EER_1 if $(EER_n/EER_1) < (1/K_1)$. In this section, we empirically validate the statement. Below, we give the experimental details.

We have four score sets available: LI, RI, C, and G. Table 4.4 shows the individual performance of the four verifiers (that generate the score sets LI, RI, C, and G) in terms of the equal error rate (EER) on the training set. According to Table 4.4, C is the best performing, RI is the next best performing, G is next best performing, and LI is the worst performing verifier. Therefore, in our experiment,

EER_1 = Equal error rate of C,

EER_4 = Equal error rate of LI, and

K_1 = Proportion of genuine scores in confusion region of C.

From Table 4.4, $EER_1 = 0.0463$ and $EER_4 = 0.0772$. Hence, (EER_4/EER_1) = (0.0772/0.0463) = 1.6674. From Table 4.2, $K_1 = 0.5792$. Hence, $(1/K_1) = (1/0.5792)$ = 1.7265. We see that $(EER_4/EER_1) < \frac{1}{K_1}$. Therefore, according to our findings in

Section 4.2, if we build a four-stage verification system such that verifier C is placed in Stage 1, verifier RI is placed in Stage 2, verifier G is placed in Stage 3, and verifier LI is placed in Stage 4, the false alarm rate and the impostor pass rate of the four-stage system will be less than the equal error rate of the top performing verifier C (*i.e.*, FAR_4 and IPR_4 will be less than EER_1).

Table 4.4: Individual performance of the four verifiers (that generate the scores in LI, RI, C, and G) in terms of equal error rate (EER) on the training set.

	Fingerprint verifiers		**Face verifiers**	
	LI	RI	C	G
EER	0.0772	0.0547	0.0463	0.0579

To validate the above statement (*i.e.*, FAR_4 and IPR_4 will be less than EER_1), we model a four-stage verification system, accordingly. Figure 4.6 presents the system, where Stage 1 uses score set C, Stage 2 uses score set RI, Stage 3 uses score set G, and Stage 4 uses score set LI. In Stages 1, 2, and 3, the score s_i is compared with two reject thresholds L_i and R_i, for $i = 1$, 2, and 3, respectively. If s_i lies in the interval $[L_i, R_i]$, the system proceeds to the next stage. In the fourth stage, the subject is classified into genuine or impostor using a single threshold T.

In Stages 1, 2, and 3, we apply the symmetric rejection method (*i.e.*, we calculate reject regions L_1R_1, L_2R_2, and L_3R_3 using the symmetric rejection method). In the symmetric rejection method, we need to specify the upper bound for the genuine reject rate, x. From the specified upper bound x, we estimate the value of α_G and use the estimated value of α_G to calculate the reject region L_iR_i by Algorithm 1. We experimented with 10 different values of x. The minimum possible value of x is zero and the maximum possible value of x is K. Table 4.2 shows the Ks of score sets LI, RI, C, and G, estimated on training data.

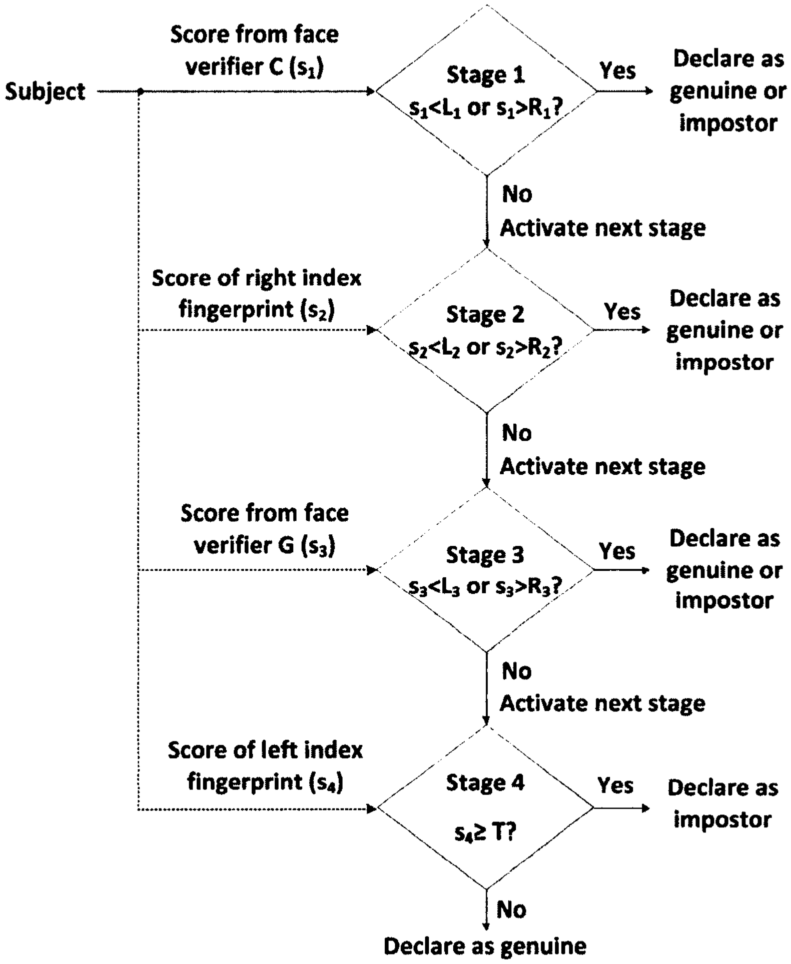

Subject
Score from face verifier C (s_1)
Stage 1
$s_1<L_1$ or $s_1>R_1$?
Yes
Declare as genuine or impostor
No
Activate next stage
Score of right index fingerprint (s_2)
Stage 2
$s_2<L_2$ or $s_2>R_2$?
Yes
Declare as genuine or impostor
No
Activate next stage
Score from face verifier G (s_3)
Stage 3
$s_3<L_3$ or $s_3>R_3$?
Yes
Declare as genuine or impostor
No
Activate next stage
Score of left index fingerprint (s_4)
Stage 4
$s_4\geq T$?
Yes
Declare as impostor
No
Declare as genuine

Figure 4.6: A four-stage biometric verification system based on the score sets LI, RI, C, and G.

We start with $x = 0.1K$. When we specify $x = 0.1K$, x is set to 0.1*0.5792 or 0.0579 in Stage 1 (verifier C), 0.1*0.1660 or 0.0166 in Stage 2 (verifier RI), and 0.1*0.4054 or 0.0405 in Stage 3 (verifier G). In other words, by specifying $x = 0.1K$, we actually specify that Stage 1 is not allowed to reject more than 5.79% of the genuine scores, Stage 2 is not allowed to reject more than 1.66% of the genuine scores, and Stage 3 is not allowed to reject more than 4.05% of the genuine scores. After initializing x to $0.1K$, we increment x 9 times, each time by $0.1K$, up to K. Table 4.5 shows the interpretation of the values of 'x' used in the experiment.

Table 4.5: Interpretation of the values of 'x' used in the experiment to generate the results in Table 4.6. For example, by specifying $x = 0.1K$, we actually specify that Stage 1 (verifier C) is not allowed to reject more than 5.79% of the genuine scores, Stage 2 (verifier RI) is not allowed to reject more than 1.66% of the genuine scores, and Stage 3 (verifier G) is not allowed to reject more than 4.05% of the genuine scores. Table 4.2 shows the Ks of verifiers C, RI, and G, estimated on training data.

Specified value of x	**Interpretation of x in Stages 1, 2, and 3**		
	x in Stage 1	**x in Stage 2**	**x in Stage 3**
$0.1K$	5.79%	1.66%	4.054%
$0.2K$	11.58%	3.32%	8.108%
$0.3K$	17.38%	4.98%	12.16%
$0.4K$	23.17%	6.64%	16.22%
$0.5K$	28.96%	8.30%	20.27%
$0.6K$	34.75%	9.96%	24.32%
$0.7K$	40.54%	11.62%	28.38%
$0.8K$	46.34%	13.28%	32.43%
$0.9K$	52.13%	14.94%	36.49%
K	57.92%	16.60%	40.54%

In Stage 4, we set the value of T to EER_4 (which is the equal error rate of LI). From Table 4.4, EER_4 is equal to 0.0772.

We present the results in Table 4.6. Table 4.6 shows that FAR_4s and IPR_4s are less than EER_1 (which is equal to 0.0463) for all values of x. The false alarm rate has been reduced by a maximum of 91.58% (when $x = 0.9K$) and the impostor pass rate has been reduced by a maximum of 56.16% (when $x = 0.7K$) compared to EER_1 (*i.e.*, compared to the equal error rate of the top performing individual verifier C). On average, the false alarm rate has been reduced by 43.05% and the impostor pass rate has been reduced by 32.68% compared to EER_1.

Table 4.6: False alarm rates and impostor pass rates of the four-stage biometric verification system presented in Figure 4.6. We experimented with 10 different values of x, where x is the specified upper bound for genuine reject rate. Table 4.5 shows the interpretation of the values of 'x' used in the experiment.

x	**False Alarm Rate (FAR_4)**	**Impostor Pass Rate (IPR_4)**
$0.1K$	0.0349	0.0424
$0.2K$	0.0349	0.0418
$0.3K$	0.0349	0.0371
$0.4K$	0.0349	0.0321
$0.5K$	0.0310	0.0279
$0.6K$	0.0310	0.0235
$0.7K$	0.0271	0.0203
$0.8K$	0.0156	0.0290
$0.9K$	0.0039	0.0252
K	0.0155	0.0324

In summary, the results of our experiment validate our finding: FAR_n and IPR_n will be less than EER_1 if $\frac{EER_n}{EER_1} < \frac{1}{K_1}$.

4.4 Empirical Comparison of Symmetric Rejection Method with Other Methods

We compared the performance of the symmetric rejection method with two existing rejection methods: (1) SPRT-based method (see [26] and [27]), which uses score-fusion and (2) Marcialis *et al.*'s method (see [23]), which does not use score-fusion. Below, we briefly describe these two methods.

SPRT-based Rejection Method: The SPRT (sequential probability ratio test) based rejection method is a sequential score-fusion method, which uses the likelihood ratio of verification score and fuses the score of i^{th} stage with the scores of the previous stages. Assuming the real valued similarity scores, the reject thresholds L and R, in the SPRT-based method, are set as follows: $L = \frac{\alpha}{1-\beta}$, $R = \frac{1-\alpha}{\beta}$, where α and β are the desired false alarm rate and the desired impostor pass rate, respectively (see [58]).

Marcialis et al.'s Rejection Method: Marcialis *et al.*'s rejection method is a non score-fusion method, *i.e.*, it does not fuse the verification score of i^{th} stage with the verification scores of the previous stages. The method is simple–it selects the whole confusion region as the reject region.

4.4.1 Experiments

We did experiments on the four-stage biometric verification system presented in Figure 4.6. In Stages 1, 2, and 3 of the verification system, we need to choose the reject regions L_1R_1, L_2R_2, and L_3R_3, respectively. Below, we describe how we chose the reject regions in three rejection methods: Symmetric, SPRT-based, and Marcialis *et al.*'s, in our experiments.

Symmetric Method: In the symmetric rejection method, we need to specify the upper bound for genuine reject rate, x. We experimented with 40 different values of x. Specifically, we set the initial value of x to $0.025K$. Then we incremented the value of x 39 times, each time by $0.025K$, up to K (which is the maximum possible value of x).

SPRT-based Method: In SPRT-based rejection method, we need to set two parameters: (1) the desired false alarm rate (α) and (2) the desired impostor pass rate (β). We experimented with equal value of α and β, which we denote as ϵ ($= \alpha = \beta$). We used the same value of ϵ in all stages of the system. We experimented with 500 different values of ϵ. Specifically, we set the initial value of ϵ to 0.001 and then incremented the value 499 times, each time by 0.001. Furthermore, SPRT-based rejection method requires the estimation of underlying probability density functions of genuine and impostor scores in order to compute the likelihood ratios. We used Gaussian mixture models (GMM) to estimate the densities because (1) it has been successfully used in several studies (for example, [9] and [26]), and (2) studies in [59] and [60] theoretically show that the density estimates produced by finite mixture models converges on the true densities if there are a sufficient number of training samples. In GMM-based density estimation, we need to set the number of components, k. After performing some preliminary experiments, we set the values of k to 2 for LI, RI, and C and 3 for G.

Marcialis et al.'s Method: Marcialis *et al.*'s rejection method chooses the whole confusion region as the reject region. Note that Marcialis *et al.*'s method is an extreme case of the symmetric rejection method. In particular, the reject region selected by Marcialis *et al.*'s method is the same as the reject region selected by the symmetric method when x is maximum (*i.e.*, when $x = K$).

In Stage 4 (which is the terminal stage), we need to set a single threshold, T. Because we compared three different rejection methods, to keep consistency in the experiment, we did not fix the value of T, but rather, we generated a receiver operating characteristic (ROC) curve by varying the value of T over the scoreline.

4.4.2 Results and Analysis

We measure the performance of the four-stage verification system, presented in Figure 4.6, using the trade-off between the area under the ROC curve (see [61], [62], and [63]) and the average number of stages required (see [26] and [27]). The area under the ROC curve (AUC) quantifies the overall ability of a verifier to discriminate between genuine users and impostors. The value of AUC for a perfect verifier, which yields zero false alarm rate and zero impostor pass rate, is 1. The value of AUC for a verifier that performs like a random guess is 0.5. Minimally, a verifier should perform better than a random guess. The higher the value of AUC, the better the verifier.

The average number of stages (ANS) required to get verification decision is directly related to the reject rate of the verification scores. Because one of our primary goals in this paper is to control the reject rate of *genuine scores* (*GRR*), we are interested in the average number of stages (ANS) required to verify the *genuine scores.* A small value of ANS required to verify the *genuine scores* indicates high user-convenience. We also report the ANS required to verify the *impostor scores* to show the full picture of the system's performance.

Trade-off between "area under ROC curve" and "average number of stages required to verify the genuine scores": Figure 4.7 shows the trade-off curves between the area under the ROC curve (AUC) and the average number of stages (ANS) required to verify the *genuine scores.* We calculated the values of AUC and ANS at 40 different values of x in the case of the symmetric rejection method and 500 different values of ϵ in the case of SPRT-based rejection method. In the symmetric rejection method, AUC and ANS increase with the increase of x and in SPRT-based method, AUC and ANS increase with the decrease of ϵ. To avoid cluttering the figures, we indicate the x and ϵ values corresponding to only some points in the plot.

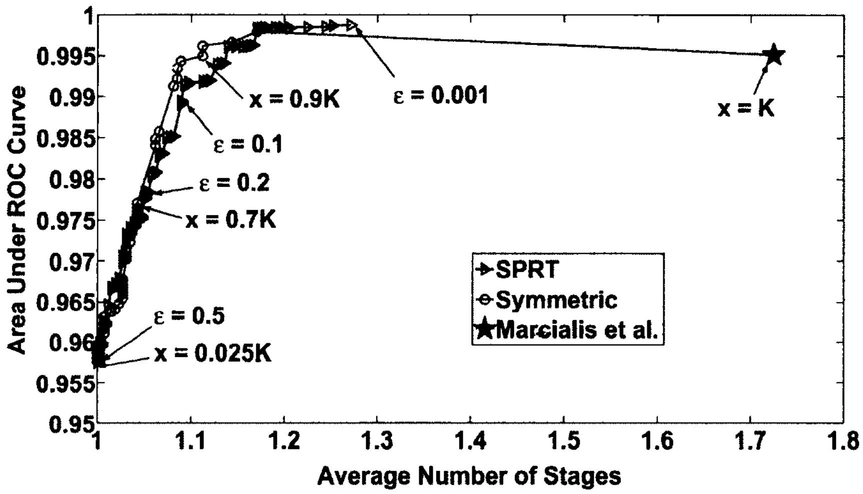


Figure 4.7: Comparing the performance of three rejection methods: Symmetric, SPRT-based, and Marcialis *et al.*'s evaluated on the verification system presented in Figure 4.6. Performance metric is–trade-off between "area under the ROC curve" and the "average number of stages required to verify the ***genuine scores***".

Our observations from Figure 4.7 are listed below.

- *Observation 1:* The symmetric rejection method performs better than the SPRT-based method in terms of trade-off between AUC and ANS required by the genuine scores. For example, to achieve AUC = 0.985, the symmetric rejection method requires ANS = 1.062 and the SPRT-based method requires ANS = 1.081. Similarly, to achieve AUC = 0.995, the symmetric rejection method requires ANS = 1.112 and the SPRT-based method requires ANS = 1.139. However, the SPRT-based method produces slightly higher AUC at the expense of higher ANS in comparison to the symmetric rejection method. For example, the maximum value of AUC achieved by the SPRT-based method is 0.999 (at the expense of ANS = 1.271 at $\epsilon = 0.001$), whereas the maximum value of AUC achieved by the symmetric rejection method is 0.998 (at the expense of ANS = 1.17 at $x = 0.975K$).

- *Observation 2:* Performance of Marcialis *et al.*'s method is worse than both the symmetric and the SPRT-based methods in terms of trade-off between AUC and ANS required by the genuine scores. AUC obtained by Marcialis *et al.*'s method is 0.995; however, it requires ANS = 1.725. To achieve the same AUC, the symmetric rejection method requires ANS = 1.112 and the SPRT-based method requires ANS = 1.139.

Summary: Figure 4.7 signifies that to achieve the same AUC, genuine users require less number of stages (which incurs less cost in terms of biometric sample data acquisition effort and verification time) in the symmetric rejection method compared to the SPRT-based rejection method. This indicates that the symmetric rejection method can provide better user convenience and administrator convenience, which are desirable attributes, especially for applications involving a large population of users or a great number of biometric transactions.

Trade-off between "area under ROC curve" and "average number of stages required to verify the impostor scores": Figure 4.8 shows the trade-off curves between AUC and ANS required to verify the *impostor scores.* Our observations from Figure 4.8 are listed below.

- *Observation 1:* The SPRT-based method performs better than the symmetric rejection method in terms of trade-off between AUC and ANS required by the impostor scores. For example, to achieve AUC = 0.976, the symmetric rejection method requires ANS = 1.124 and the SPRT-based method requires ANS =

1.068. Similarly, to achieve AUC = 0.985, the symmetric rejection method requires ANS = 1.304 and the SPRT-based method requires ANS = 1.118.

- *Observation 2:* Performance of Marcialis *et al.*'s method is worse than both the symmetric and the SPRT-based method in terms of trade-off between AUC and ANS required by the impostor scores. AUC obtained by Marcialis *et al.*'s method is 0.995; however, it requires ANS = 2.638. To achieve the same AUC, the symmetric rejection method requires ANS = 2.116 and the SPRT-based method requires ANS = 1.276.

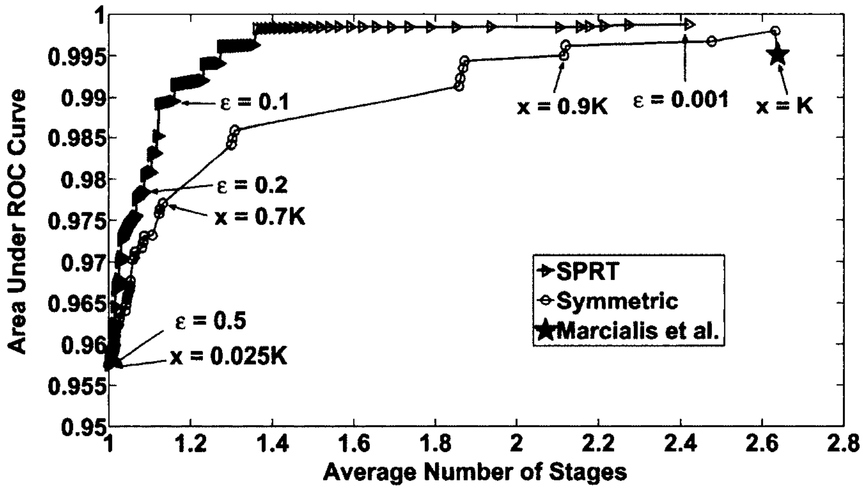


Figure 4.8: Comparing the performance of three rejection methods: Symmetric, SPRT-based, and Marcialis *et al.*'s evaluated on the verification system presented in Figure 4.6. Performance metric is–trade-off between "area under the ROC curve" and the "average number of stages required to verify the ***impostor scores***".

Summary: Figure 4.8 signifies that to achieve the same AUC, impostors require more stages in the symmetric rejection method compared to the SPRT-based rejection method. This indicates that the symmetric rejection method gives more inconvenience to the impostors (which is good); however, it incurs more cost to verify the impostors, which, in turn, increases the administrator-inconvenience (which is bad).

CHAPTER 5

PROPOSED FUSION FRAMEWORK FOR MULTI-BIOMETRIC VERIFICATION

In this chapter, we propose a new fusion framework for multi-biometric verification systems, which: (1) achieves an accuracy level higher than parallel fusion, and at the same time (2) provides convenience to genuine users. We theoretically show that the false alarm rate and impostor pass rate obtained by our proposed fusion framework is less than or equal to the false alarm rate and impostor pass rate obtained by parallel fusion. We validate our theory by experimenting on ***two*** multi-biometric verification systems.

The rest of the chapter is organized as follows: In Section 5.1, we present our proposed fusion framework and theoretically show how it (1) gives better performance than parallel fusion, and (2) provides convenience to genuine users. In Section 5.2, we give our experimental details and analyze the results.

5.1 Proposed Fusion Framework

We propose an information fusion framework for multi-biometric verification systems. Let v_1, v_2, $\cdots$, v_n be n biometric verifiers such that $EER_1 < EER_2 < \cdots < EER_n$, where EER_i is the equal error rate of verifier v_i, for $i = 1, 2, \cdots, n$. We model an $(n+1)$-stage multi-biometric verification system as follows:

(1) Verifier v_1 is placed in the first stage, verifier v_2 is placed in the second stage, and so on, until verifier v_n is placed in the n^{th} stage.

(2) Stage i, for $i = 1, 2, \cdots, n$, gives a verification decision when it is ***fully confident*** about the decision, *i.e.*, it declares a subject as genuine when it is fully confident that he/she is genuine and it declares a subject as an impostor when it is fully confident that he/she is an impostor. If Stage i is not fully confident, it does not give any verification decision; rather, it rejects the biometric sample and activates the $(i+1)^{th}$-stage. In such a case, the subject will have to submit the $(i+1)^{th}$ biometric trait to Stage $i+1$ to get the verification decision.

By "fully confident" we mean that when Stage i gives a verification decision, it ensures that no genuine user is going to be erroneously declared as an impostor and no impostor is going to be erroneously declared as a genuine user. As a result, the given verification decision incurs zero false alarm rate and zero impostor pass rate. Below, we explain how Stage i can give verification decision with full confidence.

We use Figure 5.1 to explain the operation of Stage i. We assume X and Y are genuine and the impostor score sets, respectively, are generated by the verifier v_i at Stage i; $f_G(x)$ and $f_I(y)$ are the distributions estimated from scores in X and Y, respectively. Without loss of generality, we assume the verifier v_i outputs real-valued dissimilarity scores (*e.g.*, Euclidean distance between a biometric sample and the claimed template). In the scoreline $[Z, O]$, ZE_2 is the genuine score region, E_1O is the impostor score region, and E_1E_2 is the *confusion region* where $f_G(x)$ and $f_I(y)$ overlap.

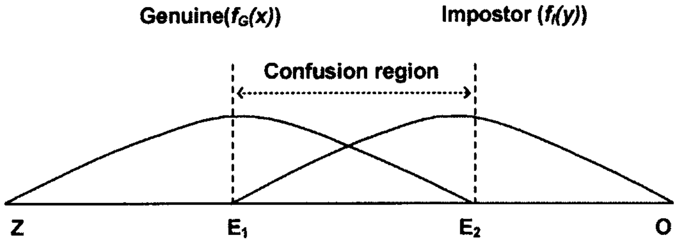


Figure 5.1: Genuine score distribution $f_G(x)$ and impostor score distribution $f_I(y)$ along with genuine score region ZE_2, impostor score region E_1O, and confusion region E_1E_2.

In Figure 5.1, the probability of a score that lies on the left side of the confusion region (*i.e.*, a score that lies inside ZE_1) to be an impostor is zero. Similarly, the probability of a score that lies on the right side of the confusion region (*i.e.*, a score that lies inside E_2O) to be genuine is zero. Hence, during verification, if a score lies on the left side of the confusion region, Stage i can be fully confident that the subject is genuine and can give verification decision accordingly. Similarly, if a score lies on the right side of the confusion region, Stage i can be fully confident that the subject is an impostor and can give verification decision accordingly. If the score lies inside the confusion region, Stage i cannot be fully confident, and hence cannot give any verification decision.

The above way of "being fully confident" while giving verification decision is theoretically sound; however, in practice, it may not work properly (*i.e.*, it may not give our expected result). Specifically, in practice, the confusion region of the testing scores may differ from the confusion region estimated on the training scores in such a way that some part of the confusion region of the testing scores lies outside the confusion region of the training scores. In such a case, the above way of "being fully confident" may lead to some wrong verification decisions. As a consequence, we may fail to reach our target of zero false alarm rate and zero impostor pass rate at Stage i.

To ensure zero false alarm rate and zero impostor pass rate at Stage i, we add some safety region to both sides of the confusion region and form a ***confident reject region***. Below, we explain the concept of confident reject region.

In Figure 5.2, we illustrate a confident reject region L_cR_c, where L_c is the left threshold and R_c is the right threshold. We calculate L_c and R_c based on the length of the confusion region E_1E_2, as follows:

$$\text{Length of } E_1E_2 = E_2 - E_1,$$

$$L_c = E_1 - p \text{ percent of the length of } E_1E_2, \text{ and}$$

$$R_c = E_2 + p \text{ percent of the length of } E_1E_2,$$

where p is a real number greater than or equal to zero; p determines the length of the safety region and hence we refer to p as the ***safety level***. In Figure 5.2, L_cE_1 is the left safety region and E_2R_c is the right safety region. The length of the left safety region is equal to the length of the right safety region.

Note that when p is equal to zero, the confident reject region is the same as the confusion region. When p is equal to 50, 50 percent of the confusion region is added to each side of the confusion region and the confident reject region becomes double the confusion region. Similarly, when p is equal to 100, each safety region becomes equal to the confusion region and the confident reject region becomes triple the confusion region. Because the confident reject region changes as the value of p changes, p controls the performance of the proposed framework. An optimal value of p, which gives the minimum verification error, can be estimated using well-known parameter selection techniques such as cross-validation on a hold-out dataset, bootstrapping [64], or genetic algorithm [65].

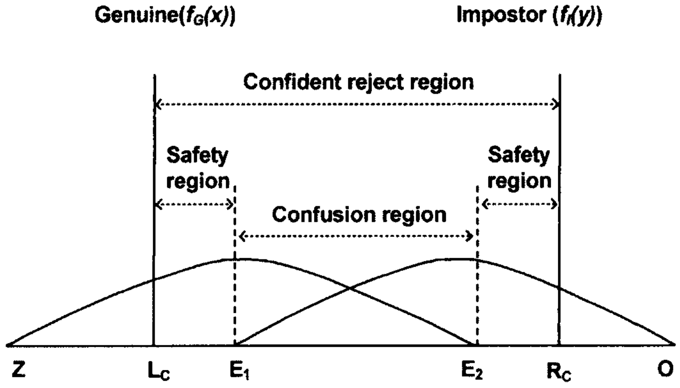


Figure 5.2: Illustration of the confident reject region. E_1E_2 is the confusion region, L_cE_1 is the left safety region, E_2R_c is the right safety region, L_cR_c is the confident reject region, ZL_c is the confident genuine region, and R_cO is the confident impostor region.

Now we are ready to formally state the biometric verification rule with the confident reject region. Let s denote a score output by verifier v_i at Stage i when a biometric sample S is matched with a claimed template C. Assuming v_i outputs a dissimilarity score (*e.g.*, Euclidean distance between features of S and C), biometric verification with the confident reject region L_cR_c is implemented as follows:

S is *genuine* if $s < L_c$, *impostor* if $s > R_c$, else reject S.

In Figure 5.2, we refer to ZL_c as the confident genuine region and R_cO as the confident impostor region.

(3) Stage $n + 1$ gives verification decision to those subjects who fail to get the decision in stages 1 through n. Stage $n + 1$ uses a parallel fusion approach

(*e.g.*, arithmetic combination, classification, or density based approach) to make the verification decision.

Example: In Figure 5.3, we illustrate the working principle of our proposed fusion framework. We have two biometric verifiers: (1) a face verifier and (2) a fingerprint verifier. Let the equal error rate of the face verifier be less than the equal error rate of the fingerprint verifier. We model a three-stage verification system as follows: we place the face verifier in Stage 1, the fingerprint verifier in Stage 2, and use the weighted sum fusion of face and fingerprint in Stage 3. The weighted sum fusion [1] is a well-known parallel fusion method.

To verify a subject, we submit his/her face sample to the face verifier at Stage 1. The face verifier matches the sample with the claimed template and generates a score x_1. The score x_1 is then compared with two thresholds $L_{\mathrm{c},1}$ and $R_{\mathrm{c},1}$, where $L_{\mathrm{c},1}R_{\mathrm{c},1}$ is the confident reject region of the face score distribution. If x_1 is less than $L_{\mathrm{c},1}$, the subject is declared as a genuine user, if x_1 is greater than $R_{\mathrm{c},1}$, the subject is declared as an impostor, and if x_1 lies inside $[L_{c,1},\ L_{c,1}]$, the face sample is rejected without giving any verification decision. When the face sample is rejected by Stage 1, we activate Stage 2, submit a fingerprint sample of the subject to the fingerprint verifier at Stage 2, and follow the same procedure. In Figure 5.3, x_2 denote the matching score generated by the fingerprint verifier and $L_{c,2}R_{c,2}$ denote the confident reject region of the fingerprint score distribution.

If the fingerprint sample is rejected by Stage 2, we activate Stage 3. In Stage 3, we perform the weighted sum fusion over the scores x_1 and x_2. Specifically, we calculate the fused score $x_f = w_1x_1 + w_2x_2$ and compare x_f with a threshold T. If x_f

is less than T, the subject is declared as a genuine user; otherwise, he/she is declared as an impostor.

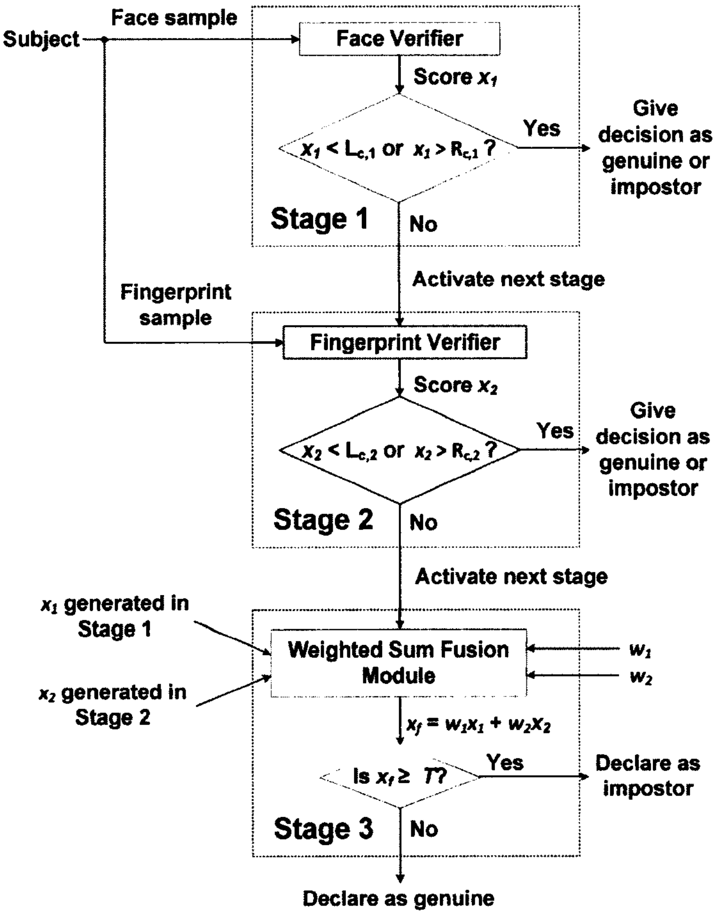


Figure 5.3: Illustration of our proposed fusion framework using a three-stage verification system, where x_1 denotes the matching score generated by the face verifier, x_2 denotes the matching score generated by the fingerprint verifier, $L_{c,1}R_{c,1}$ denotes the confident reject region of the face score distribution, and $L_{c,2}R_{c,2}$ denotes the confident reject region of the fingerprint score distribution.

Note on Verifier Order: The order in which the individual verifiers are placed in a multi-stage verification system is called *verifier order*. Because we use a confident reject region, which ensures zero false alarm rate and zero impostor pass rate in stages 1 through n of an $(n + 1)$-stage verification system, the verifier order has theoretically no effect on the verification errors. However, in practice, there is always some chance of intrusion in the case of weak biometric verifiers. Hence, if we allow the users to decide which biometric trait to submit first, a clever impostor may use the weakest biometric verifier first and may get access to the system. To prevent such security breaches, in our proposed fusion framework, we suggested the following verifier order: the best performing individual verifier in the first stage, the next best performing individual verifier in the second stage, and so on, until the worst performing individual verifier in the n^{th} stage of an $(n+1)$-stage verification system. Use of this verifier order makes our proposed framework applicable to high security applications.

5.1.1 Performance Improvement by the Proposed Fusion Framework

In Figure 5.4, we give a two-unit representation of our proposed fusion framework, where the ***confident unit*** consists of stages 1 through n and the ***parallel unit*** consists of the $(n+1)^{th}$ stage of an $(n+1)$-stage verification system. Because each stage in the confident unit gives verification decision when it is fully confident, the false alarm rate and impostor pass rate incurred by the confident unit is zero. Therefore, the only source of verification error in our proposed framework is the parallel unit. This phenomenon enables the proposed fusion framework to give better performance than parallel fusion framework in terms of impostor pass rate and false alarm rate.

Below, we prove that the false alarm rate obtained by our proposed fusion framework is less than or equal to the false alarm rate obtained by the parallel fusion framework. The proof that the impostor pass rate obtained by our proposed fusion framework is less than or equal to the impostor pass rate obtained by the parallel fusion framework is similar and hence it is given in Appendix C.

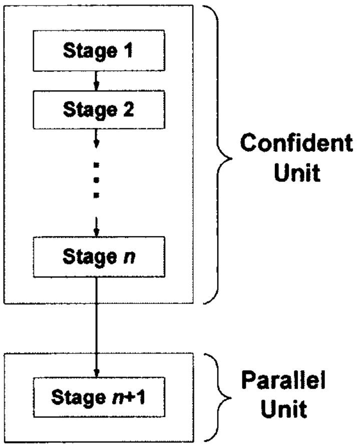


Figure 5.4: A two-unit representation of the proposed fusion framework.

Proof that the False Alarm Rate Obtained by Our Proposed Fusion Framework is Less than or Equal to the False Alarm Rate Obtained by the Parallel Fusion Framework: For simplicity, we prove the statement using a specific number of genuine users (the proof is similar for any number of genuine users). We use Figure 5.5 and Figure 5.6 to explain the proof.

Let G_1, G_2, G_3, G_4, G_5, G_6, G_7, G_8, G_9, and G_{10} be 10 genuine users. First, we calculate the false alarm rate obtained by the proposed fusion framework. We use Figure 5.5 in this regard. When we apply the proposed fusion framework, let genuine users G_1, G_2, G_3, G_4, and G_5 receive verification decision by the confident unit and the remaining genuine users G_6, G_7, G_8, G_9, and G_{10} receive verification decision by the parallel unit of the framework. We define two groups of genuine users on the basis of the verification decisions the proposed framework gives:

- Group I: The genuine users who receive verification decision by the confident unit. In Figure 5.5, Group I consists of genuine users G_1, G_2, G_3, G_4, and G_5.
- Group II: The genuine users who receive verification decision by the parallel unit. In Figure 5.5, Group II consists of the genuine users G_6, G_7, G_8, G_9, and G_{10}.

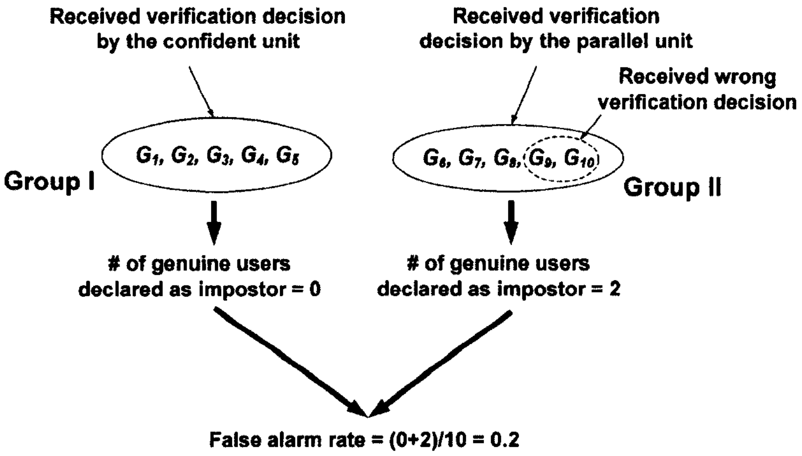

Received verification decision by the confident unit
Received verification decision by the parallel unit
Received wrong verification decision
Group I
G1, G2, G3, G4, G5
G6, G7, G8, G9, G10
Group II
# of genuine users declared as impostor = 0
# of genuine users declared as impostor = 2
False alarm rate = (0+2)/10 = 0.2

Figure 5.5: Calculation of false alarm rate obtained by the proposed fusion framework.

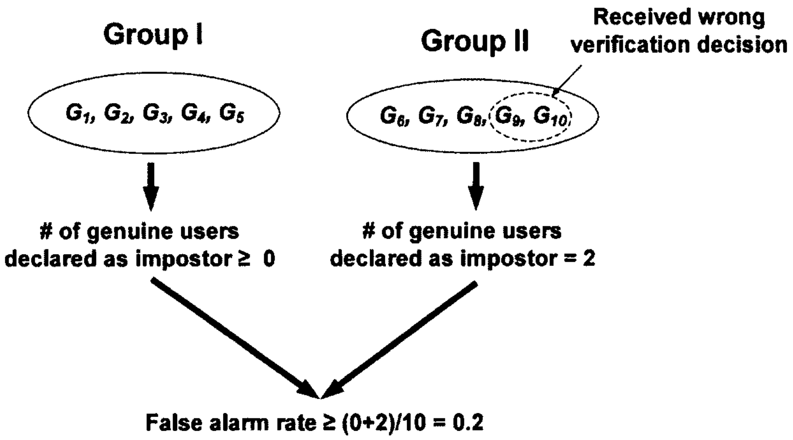

Group I
Group II
Received wrong verification decision
G1, G2, G3, G4, G5
G6, G7, G8, G9, G10
# of genuine users declared as impostor ≥ 0
# of genuine users declared as impostor = 2
False alarm rate ≥ (0+2)/10 = 0.2

Figure 5.6: Calculation of false alarm rate obtained by the parallel fusion framework.

Because the confident unit incurs zero false alarm rate, the number of genuine users in Group I erroneously declared as impostor is zero. However, because the parallel unit may give some wrong verification decisions, the number of genuine users in Group II erroneously declared as impostor is greater than or equal to zero. Let the parallel unit declare the genuine users G_9 and G_{10} as impostor (for clarity, G_9 and G_{10} are enclosed by a dotted circle in Figure 5.5). Hence, the total number of genuine users declared as impostor by the proposed fusion framework is *equal* to (0+2) or 2. Therefore, the false alarm rate obtained by the proposed fusion framework is *equal* to 2/10 or 0.2.

Now, we calculate the false alarm rate obtained by the parallel fusion framework. We use Figure 5.6 in this regard.

In Figure 5.6, we apply parallel fusion separately on Group I and Group II. Because parallel fusion may give some wrong verification decisions, when we apply parallel fusion on Group I, the number of genuine users in Group I erroneously declared as impostor is greater than or equal to zero. When we apply parallel fusion on Group II, we achieve exactly the same result as we achieve by applying the parallel unit of the proposed framework on Group II, *i.e.*, the genuine users G_9 and G_{10} receive the wrong verification decision. Hence, the total number of genuine users declared as impostor by the parallel fusion framework is *greater than or equal* to 2. Therefore, the false alarm rate obtained by the parallel fusion framework is greater than or equal to 2/10, *i.e.*, *greater than or equal* to 0.2.

5.1.2 Convenience to Genuine Users Provided by the Proposed Fusion Framework

The proposed fusion framework provides convenience to genuine users by allowing them to submit a subset of the biometric traits. For example, if a user receives verification decision at the first stage, he/she does not need to submit other biometric traits. Thus, our proposed fusion framework can save a lot of time for the genuine users. In contrast, in the parallel fusion framework, a person needs to submit all of the n biometric traits in case of an n-biometric verification system. As a result, the verification task takes a lot of time. The problem becomes severe in the applications that involve a large population of users or a huge number of biometric transactions. Obviously, our proposed fusion framework can be an effective solution to this problem.

Note that while the proposed fusion framework can provide a considerable amount of convenience to a large number of genuine users, a few genuine users may have to go through the parallel unit, and they will require a bit more time than the time required to be verified by a parallel fusion framework. The good news is the time difference is so small that we should not be worried. Let us explain in detail. We have two biometric verifiers available: a face verifier and a fingerprint verifier. In parallel fusion, *e.g.*, in weighted sum fusion, the verification time for a user consists of (1) time required to submit two biometric traits (face and fingerprint), (2) time required to generate two matching scores, (3) time required to fuse the matching scores, and (4) time required to apply a threshold on the fused score. In contrast, in the proposed fusion framework (with weighted sum fusion in the parallel unit), the verification time for a user *who goes through the parallel unit* consists of (1) time required to submit two biometric traits (face and fingerprint), (2) time required to generate two matching scores, (3) time required to fuse the matching scores, and (4) time required to apply *three* thresholds in three different stages (one threshold in each stage). Therefore, a user who goes through the parallel unit needs two extra thresholdings in comparison to the parallel fusion framework. To perform a thresholding task (*i.e.*, to compare two real numbers), a personal computer with 1 GHz processor takes less than 0.1 micro seconds [66], [67]. Thus, the time difference is insignificant (in comparison to the time required for acquiring biometric sample). With this small sacrifice, the proposed fusion framework can achieve the minimum EER, and at the same time, it can provide convenience to a large population of genuine users. Hence, the proposed fusion framework can be very useful in real world biometric applications.

5.2 Empirical Validation

In this section, we empirically show that the proposed fusion framework can give better performance (in terms of equal error rate) than parallel fusion framework, and at the same time, it can provide a considerable amount of convenience to the genuine users. We did our experiments on ***two*** multi-biometric verification systems to provide substantial evidence. Below, we give the experimental details.

5.2.1 Data

We performed experiments using NIST Biometric Scores Set Release 1 (BSSR 1) [41]. The NIST-BSSR1 database consists of three score sets–1) fingerprint-face, 2) fingerprint, and 3) face. We experimented with score set fingerprint-face. The fingerprint-face set consists of face and fingerprint scores from the same set of 517 individuals. For each individual, the set contains one score from the comparison of two right index fingerprints, one score from the comparison of two left index fingerprints, and two scores (from two separate face verifiers, namely, C and G) from the comparison of two frontal faces. The scores of the right and left index fingerprints were generated using the same fingerprint verifier. The fingerprint images and the face images used to compute the scores were collected from the same person and at the same time. Hence, the fingerprint-face score set is true multi-modal.

5.2.2 Design of Experiments

We divided the fingerprint-face score set into four disjoint subsets–(1) LI: set of scores of left index fingerprints, (2) RI: set of scores of right index fingerprints, (3) C: set of scores from face verifier C, and (4) G: set of scores from face verifier G.

We experimented on 10 different training-testing sets. Each training-testing set was generated as follows: the scores of 259 individuals were randomly selected to form the training set and the rest of the scores from the 258 individuals were used to form the testing set. Table 5.1 shows the number of genuine and impostor scores in each of the 10 training-testing sets of LI, RI, C, and G.

Table 5.1: The number of genuine and impostor scores in a training-testing set of LI, RI, C, and G.

	Score sets (LI/RI/C/G)	
	Training sets	Testing sets
Genuine scores	259*1	258*1
Impostor scores	259*516	258*516

We modeled two four-stage multi-biometric verification systems based on our proposed fusion framework:

1. ***Verification System RI-G-LI:*** This system uses score set RI in the first stage, G in the second stage, and LI in the third stage. In the fourth stage, it uses weighted sum fusion [1].
2. ***Verification System C-RI-LI:*** This system uses score set C in the first stage, RI in the second stage, and LI in the third stage. In the fourth stage, it uses weighted sum fusion [1].

We did not use score sets C and G in the same verification system because C and G are generated from the same biometric trait: face. We selected the verifier order in each verification system based on the individual performance of the four verifiers (that generate the score sets LI, RI, C, and G). Table 5.2 shows the individual performance of the four verifiers in terms of percentage equal error rate (% EER) estimated on testing scores of the 10 training-testing sets. The "Average" column gives the means of percentage EERs calculated over the 10 sets. On the basis of the average of percentage EERs, C is the best performing individual verifier and LI is the worst performing individual verifier. In both verification systems (RI-G-LI and C-RI-LI), we placed the best performing individual verifier in the first stage, the next best in the second stage, and the worst one in the third stage.

We compared: (1) the EER achieved with verification system RI-G-LI to the EER achieved with the weighted sum fusion of RI, G, and LI, and (2) the EER achieved with verification system C-RI-LI to the EER achieved with the weighted sum fusion of C, RI, and LI. In addition, we studied how much convenience the verification systems RI-G-LI and C-RI-LI can provide to the genuine users.

In our experiments, as a parallel fusion method, we used the weighted sum fusion because it is one of the top performing parallel fusion methods [1], [34].

We tested our proposed fusion framework with 11 different safety levels: 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50. Recall that when the safety level is equal to zero, the confident reject region is the same as the confusion region. When the safety level is equal to 50, the confident reject region expands by 50 percent of the length of the confusion region on each side and becomes double the confusion region.

In the weighted sum fusion, we need to choose a score normalization method. In our experiments, we chose the min-max normalization [1] because it is easy to implement and studies in [9] show that the min-max normalization is the best for NIST-BSSR1 database [41]. In the weighted sum fusion, we also need to select an appropriate weight combination. We experimented with 19 different weight combinations and reported the average of EERs obtained with them. We selected 19 weight combinations as follows: Let w_1, w_2, and w_3 be the weights assigned to three verifiers (*e.g.*, RI, G, and LI). We varied w_1, w_2, and w_3 over the range [0.25, 0.45] in steps of 0.05, such that the constraint $w_1 + w_2 + w_3 = 1$ is satisfied. Another way of explaining is we define a universal set of weights, $U_w = \{0.25, 0.3, 0.35, 0.4, 0.45\}$. Then we generate a weight combination, $W = \{w_1, w_2, w_3\}$ by taking three weights (*i.e.*, permutations with repetition) from the universal set U_w, such that the constraint $w_1 + w_2 + w_3 = 1$ is satisfied. In this way, 18 different weight combinations were generated. We also experimented with the special weight combination $\{1/3, 1/3, 1/3\}$, *i.e.*, $w_1 = w_2 = w_3$.

Table 5.2: Individual performance of the four verifiers (that generate the score sets LI, RI, C, and G) in terms of percentage equal error rate (% EER) estimated on testing scores of the 10 training-testing sets. The "Average" column gives the means of percentage EERs calculated over the 10 sets.

Verifier	**Set 1**	**Set 2**	**Set 3**	**Set 4**	**Set 5**	**Set 6**	**Set 7**	**Set 8**	**Set 9**	**Set 10**	**Average**
LI	8.5271	6.9767	9.6899	9.6899	8.5271	8.1395	8.5271	8.1395	8.9147	9.6899	**8.6821**
RI	5.693	3.876	4.6512	5.814	5.4264	5.4264	5.5871	5.5503	5.548	5.4264	**5.2999**
C	4.2636	3.1008	4.8202	3.4884	4.2388	4.078	4.6512	4.2478	5.0388	4.6512	**4.2579**
G	5.814	5.0388	5.0388	5.9304	7.3643	5.0388	5.4264	5.4241	5.5758	6.5891	**5.7241**

5.2.3 Results with Verification System RI-G-LI

Table 5.3 shows the percentage equal error rates (% EERs)[1] for 10 training-testing sets obtained by the weighted sum fusion of RI, G, and LI. In the "Average" column, we give the mean of percentage EERs calculated over the 10 sets.

Table 5.4 shows the percentage equal error rates (% EERs)[2] for 10 training-testing sets obtained by the verification system RI-G-LI when safety level (p) is varied between 0 and 50. In the "Average" column, we give the mean of percentage EERs calculated over the 10 sets.

In Table 5.4, we indicate the minimum percentage EERs for each training-testing set obtained by the verification system RI-G-LI in bold typeface. For example, in the case of sets 1, 2, 8, and 10, the verification system RI-G-LI achieves the minimum EERs at safety level zero, in the case of set 6, it achieves the minimum EER at safety level 5, in the case of sets 3, 4, 5, and 7, it achieves the minimum EERs at safety level 10, etc. If we compare the minimum EERs obtained by the verification system RI-G-LI for sets 1 to 10 (given in Table 5.4), with the EERs obtained by the weighted sum fusion of RI, G, and LI for sets 1 to 10 (given in Table 5.3), respectively, we find the following: in the case of sets 1, 2, 4, 5, 6, 7, 9, and 10, the minimum EERs achieved by the verification system RI-G-LI are less than the EERs obtained by the weighted sum fusion of RI, G, and LI (for example, in the case of set 1, the verification system RI-G-LI achieves the minimum EER 0.7752 percent, which is 14.69 percent less than the EER obtained by the weighted sum fusion of RI, G, and LI, which is

[1] For each training-testing set, we obtained percentage EERs with 19 different weight combinations and reported the mean of them.

[2] See footnote 1

0.9087 percent). For the other two sets (3 and 8), the minimum EERs achieved by the verification system RI-G-LI are equal to the EERs obtained by the weighted sum fusion of RI, G, and LI. In summary, the experiments on every training-testing set show that the verification system RI-G-LI can achieve an EER that is less than or equal to the EER obtained by the weighted sum fusion of RI, G, and LI.

In Figure 5.7, we plot the average percentage EERs at different safety levels (p), achieved with verification system RI-G-LI (given in Table 5.4, column "Average") and draw a horizontal line presenting the average percentage EER achieved with the weighted sum fusion of RI, G, and LI (given in Table 5.3, column "Average").

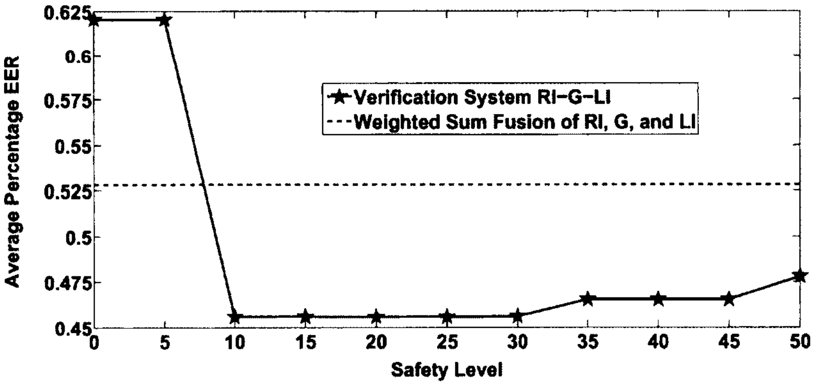


Figure 5.7: Comparing the performance of verification system RI-G-LI with weighted sum fusion of RI, G, and LI.

Our observations from Figure 5.7 are listed below:

- ***Observation 1:*** Figure 5.7 shows that at safety level zero, the average EER achieved by the verification system RI-G-LI is higher than the average EER achieved by the weighted sum fusion of RI, G, and LI. The actual fact is that at safety level zero, the EER achieved by the proposed fusion framework can be less than (see Tables 5.3 and 5.4, columns Set 1, Set 2, Set 6, and Set 10), equal

Table 5.3: Percentage equal error rates (% EERs) for 10 training-testing sets, obtained by weighted sum fusion of RI, G, and LI. The "Average" column gives the mean of percentage EERs calculated over the 10 sets.

Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 9	Set 10	Average
0.9087	0.4606	0.3876	0.4608	0.9161	0.4317	0.1929	0.3876	0.2301	0.9052	**0.5281**

Table 5.4: Percentage equal error rates (% EERs) for 10 training-testing sets, obtained by the verification system RI-G-LI when safety level (p) is varied between 0 and 50. The "Average" column gives the means of percentage EERs calculated over the 10 sets.

Safety Level	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 9	Set 10	Average
0	**0.7752**	**0.3876**	0.7752	0.7752	1.1628	0.391	0.3876	**0.3876**	0.3876	**0.7752**	0.6205
5	0.7752	0.3876	0.7752	0.7752	1.1628	**0.3909**	0.3876	0.3876	0.3876	0.7752	0.6205
10	0.7752	0.3876	**0.3876**	**0.3876**	**0.7752**	0.3909	**0.0996**	0.3876	0.1957	0.7752	0.4562
15	0.7752	0.3876	0.3876	0.3876	0.7752	0.3909	0.0996	0.3876	0.1954	0.7752	0.4562
20	0.7752	0.3876	0.3876	0.3876	0.7752	0.3909	0.0996	0.3876	**0.1947**	0.7752	**0.4561**
25	0.7752	0.3876	0.3876	0.3876	0.7752	0.3909	0.0996	0.3876	0.1947	0.7752	0.4561
30	0.7752	0.3876	0.3876	0.3876	0.7752	0.3909	0.1029	0.3876	0.1947	0.7752	0.4564
35	0.7752	0.3876	0.3876	0.3876	0.7752	0.3909	0.1929	0.3876	0.1947	0.7752	0.4654
40	0.7752	0.3876	0.3876	0.3876	0.7752	0.3909	0.1929	0.3876	0.1947	0.7752	0.4654
45	0.7752	0.3876	0.3876	0.3876	0.7752	0.3909	0.1929	0.3876	0.1947	0.7752	0.4654
50	0.816	0.3876	0.3876	0.4284	0.816	0.3909	0.1929	0.3876	0.1985	0.7752	0.4781

to (see Tables 5.3 and 5.4, column Set 8), or higher than (see Tables 5.3 and 5.4, columns Set 3, Set 4, Set 5, Set 7, and Set 9) the EER achieved by parallel fusion. Let us explain the fact.

At safety level zero, the confusion region estimated on the training scores is considered as the confident reject region. During testing, two cases can occur: (1) some part of the confusion region of the testing scores may lie outside the confusion region estimated on the training scores. In this case, the confident unit of the proposed framework will make some wrong verification decisions, and as a consequence, the EER achieved by the proposed framework may be higher than the EER achieved by parallel fusion, and (2) the confusion region of testing scores may lie inside the confusion region estimated on the training scores. In this case, the confident unit will not make any wrong verification decision, and hence, the EER achieved by the proposed framework will be less than or equal to the EER achieved by parallel fusion. In summary, it is not safe to use the confusion region as the confident reject region. We should add some safety region to the confusion region to form a confident reject region.

- ***Observation 2:*** Figure 5.7 shows that when the safety level increases from zero, the average EER achieved by the verification system RI-G-LI monotonically decreases, and at one safety level, it reaches the minimum value (0.4561 percent at safety level 20). This behavior of the proposed framework is expected because when we increase the safety level from zero, the confident reject region becomes larger than the confusion region estimated on the training scores, by expanding on both sides. As a result, the number of wrong verification decision given by the confident unit decreases, which in turn, decreases the EER achieved by the (whole) framework. As we keep increasing the safety level, at one point, we obtain our desired confident reject region which gives no wrong verification decision, and as a consequence, we achieve the minimum EER. The confident reject region that gives the minimum EER is referred to as the **optimal confident reject region**.

- ***Observation 3:*** Figure 5.7 shows that after reaching the minimum value 0.4561 percent at safety level 20, the average EER achieved by the verification system RI-G-LI monotonically increases with an increase in the safety level. The most important thing to notice is, though the EER achieved by the verification system RI-G-LI increases, it does not cross the EER achieved by the weighted sum fusion of RI, G, and LI. For example, at safety level 50, the average EER achieved by the verification system RI-G-LI reaches 0.4781, which is 9.47 percent less than the average EER achieved by the weighted sum fusion of RI, G, and LI (which is 0.5281). Let us explain the fact.

  After achieving the minimum EER, if we increase the safety level more, the confident reject region will expand more, and in turn, it will reject more scores. As a result, the confident unit will give verification decision to a less number of scores and the parallel unit will have more scores (compared to the number of scores at the safety level which gives the minimum EER) to give a verification decision. Therefore, with an increase in the safety level, the EER obtained by the proposed fusion framework converges to the EER obtained by parallel fusion. In the extreme case, with a very high safety level, the confident reject region may become so big that it will reject every score, *i.e.*, the confident unit will not give any verification decision and the parallel unit will have to handle all scores. As a result, the EER obtained by the proposed framework will be exactly same as the EER obtained by parallel fusion. Thus, after reaching the minimum value, the EER obtained by the proposed framework always remains less than or equal to the EER achieved by parallel fusion.

Performance Improvement: Table 5.5 shows the percentage of ***reduction*** in the average EER achieved with the verification system RI-G-LI (given in Table 5.4, column "Average") in comparison to the average EER achieved with weighted sum fusion of RI, G, and LI (given in Table 5.3, column "Average"). We see that verification system RI-G-LI achieved a maximum of 13.63 percent reduction in the average EER over the weighted sum fusion of RI, G, and LI. Thus, the proposed fusion framework shows considerable promise in improving the performance of multi-biometric verification systems. In Table 5.5, the percentage reductions in the average EER at safety levels 0 through 5 are negative because the average EERs obtained by the verification system RI-G-LI at safety levels 0 through 5 (given in Table 5.4, column "Average") are higher than the average EER obtained by weighted sum fusion of RI, G, and LI (given in Table 5.3, column "Average"). We explained this fact in observation 1 in this section, Page 77.

Table 5.5: Percentage of ***reduction*** in the average EER achieved with the verification system RI-G-LI (given in Table 5.4, column "Average") in comparison to the average EER achieved with the weighted sum fusion of RI, G, and LI (given in Table 5.3, column "Average"). The safety level (p) is varied between 0 and 50.

Safety Level (p)	**% Reduction in Average EER**
0	-17.4967
5	-17.4967
10	13.6148
15	13.6148
20	**13.6338**
25	13.6338
30	13.577
35	11.8728
40	11.8728
45	11.8728
50	9.4679

Table 5.6 shows the percentage of reduction in the average EER achieved with the verification system RI-G-LI in comparison to the average EER achieved with the best performing individual verifier C (given in Table 5.2, column "Average"). Specifically, we compared the minimum of the average EERs obtained by the verification system RI-G-LI (0.4561 percent) with the average EER obtained by the best individual verifier C (4.2579 percent). We observe that the proposed fusion framework (verification system RI-G-LI) reduces the equal error rate significantly (89.2881 percent).

Convenience to Genuine Users: Table 5.7 shows the percentage of genuine users getting verification decisions at different stages of the four-stage multi-biometric verification system RI-G-LI when the safety level is varied between 0 and

50.[3] In other words, Table 5.7 shows how much convenience our proposed framework (with verification system RI-G-LI) provides to the genuine users. Let us give an example. With verification system RI-G-LI, we achieved the minimum equal error rate (EER) 0.4561 percent at a safety level of 20, which is 13.63 percent less than the EER obtained with the weighted sum fusion of RI, G, and LI (0.5281 percent). Now we will see how much convenience we achieve with the verification system RI-G-LI at the safety level of 20.

Table 5.6: Percentage of reduction in the average EER achieved with verification system RI-G-LI in comparison to the average EER achieved with the best performing individual verifier C (given in Table 5.2, column "Average").

Verification System	**% Reduction in Average EER**
RI-G-LI	89.2881

In Table 5.7, we see that at the safety level of 20, 77.29 percent of the genuine users received verification decision by using only one biometric trait (RI), *i.e.*, 77.29 percent of the genuine users did not need to submit a second or third biometric trait. Also, 7.36 percent of the genuine users received verification decision by using only two biometric traits (RI and G), *i.e.*, (77.29 + 7.36) or 84.65 percent of the genuine users received verification decisions without submitting the third biometric trait (LI). In contrast, in the case of the weighted sum fusion, a user is bound to submit all of the three biometric traits (RI, G, LI). Thus, the proposed fusion framework gives the minimum EER, and at the same time, it provides a considerable amount of

[3] The percentage of genuine users getting verification decision at each stage of the verification system reported in Table 5.7 is the mean of the percentages of genuine users getting verification decisions at the corresponding stage, calculated over the 10 training-testing sets.

convenience to the genuine users. Hence, we call our proposed framework *optimized fusion framework* for multi-biometric verification systems.

Table 5.7: Percentage of genuine users getting verification decisions at different stages of the four-stage multi-biometric verification system RI-G-LI, when safety level is varied between 0 and 50.

Safety Level (p)	**1st Stage (RI)**	**2nd Stage (G)**	**3rd Stage (LI)**	**4th Stage (Parallel Unit)**
0	82.52	11.32	2.17	3.99
5	81.59	10.12	3.18	5.12
10	80.08	9.81	4.03	6.09
15	78.37	8.68	4.96	7.98
20	**77.29**	**7.36**	**5.62**	**9.73**
25	76.28	6.16	5.89	11.67
30	74.69	5.47	6.59	13.26
35	72.83	3.49	7.4	16.28
40	70.74	2.44	8.53	18.29
45	69.61	1.74	8.95	19.69
50	67.48	0.93	9.65	21.94

Note–At the safety level of 20, 9.73 percent of the genuine users received verification decision by the fourth stage of the verification system RI-G-LI, *i.e.*, by the parallel unit of the proposed framework and hence they should have required more time (to be verified) than the weighted sum fusion of RI, G, and LI. In Section 5.1.2, we discussed this fact and showed that this time difference is insignificant. With this small sacrifice, the proposed framework achieved the minimum EER, and at the same time, it provided convenience to a large population of genuine users.

5.2.4 Results with Verification System C-RI-LI

Table 5.8 shows the percentage equal error rates (% EERs) for 10 training-testing sets obtained by the weighted sum fusion of C, RI, and LI. In the "Average" column, we give the mean of percentage EERs calculated over the 10 sets.

Table 5.9 shows the percentage equal error rates (% EERs) for 10 training-testing sets obtained by the verification system C-RI-LI when the safety level (p) is varied between 0 and 50. In the "Average" column, we give the means of percentage EERs calculated over the 10 sets.

In Table 5.9, we indicate the minimum percentage EERs for each training-testing set obtained by the verification system C-RI-LI in bold typeface. For example, in the case of sets 1, 2, and 8, the verification system C-RI-LI achieves the minimum EERs at safety level zero, in the case of set 6, it achieves the minimum EER at safety level 5, in the case of sets 3, 5, and 7, it achieves the minimum EERs at a safety level of 10, etc. If we compare the minimum EERs obtained by the verification system C-RI-LI for sets 1 to 10 (given in Table 5.9), with the EERs obtained by the weighted sum fusion of C, RI, and LI for sets 1 to 10 (given in Table 5.8), respectively, we find that in the case of sets 1, 2, 3, 4, 6, 8, and 9, the minimum EERs achieved by the verification system C-RI-LI are less than the EERs obtained by the weighted sum fusion of C, RI, and LI (for example, in the case of set 1, the verification system C-RI-LI achieves the minimum EER 0.1887 percent, which is 34.46 percent less than the EER obtained by weighted sum fusion of C, RI, and LI, which is 0.2879 percent). For the other three sets (5, 7, and 10), the minimum EERs achieved by the verification system C-RI-LI are equal to the EERs obtained by the weighted sum fusion of C, RI, and LI. In summary, experiments on every training-testing set show that the verification system C-RI-LI can achieve an EER that is less than or equal to the EER obtained by the weighted sum fusion of C, RI, and LI.

In Figure 5.8, we plot the average percentage EERs at different safety levels, achieved with verification system C-RI-LI (given in Table 5.9, column "Average") and draw a horizontal line presenting the average percentage EER achieved with the weighted sum fusion of C, RI, and LI (given in Table 5.8, column "Average").

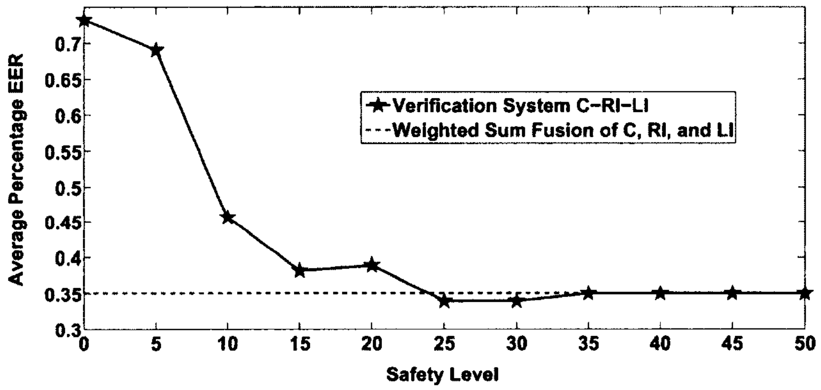


Figure 5.8: Comparing the performance of verification system C-RI-LI with weighted sum fusion of C, RI, and LI.

Our observations from Figure 5.8 are listed below:

- ***Observation 1:*** At safety level zero, the average EER achieved by the verification system C-RI-LI is higher than the average EER achieved by the weighted sum fusion.
- ***Observation 2:*** When the safety level increases from zero, the average EER achieved by the verification system C-RI-LI decreases (with one small exception at safety level 20), and at one safety level, it reaches the minimum value (0.3399 percent at safety level 25).

Table 5.8: Percentage equal error rates (% EERs) for 10 training-testing sets, obtained by weighted sum fusion of C, RI, and LI. The "Average" column gives the mean of percentage EERs calculated over the 10 sets.

Set 1	**Set 2**	**Set 3**	**Set 4**	**Set 5**	**Set 6**	**Set 7**	**Set 8**	**Set 9**	**Set 10**	**Average**
0.2879	0.2938	0.3888	0.2036	0.533	0.2607	0.2637	0.4476	0.5185	0.3095	**0.3507**

Table 5.9: Percentage equal error rates (% EERs) for 10 training-testing sets, obtained by the verification system C-RI-LI when safety level (p) is varied between 0 and 50. The "Average" column gives the means of percentage EERs calculated over the 10 sets.

Safety Level	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 9	Set 10	Average
0	**0.1887**	**0.2899**	0.8083	0.7752	0.9145	0.3876	0.8425	**0.3876**	1.1789	1.5504	0.7324
5	0.2893	0.292	0.819	0.7752	0.9306	**0.2287**	0.844	0.3876	1.1808	1.1628	0.691
10	0.288	0.294	**0.3169**	0.2034	**0.533**	0.2287	**0.2637**	0.3876	0.8878	1.1628	0.4566
15	0.288	0.2938	0.3171	**0.2028**	0.5334	0.2287	0.2637	0.3876	**0.503**	0.8043	0.3822
20	0.288	0.2942	0.3888	0.2028	0.533	0.2295	0.2637	0.3876	0.503	0.8041	0.3895
25	0.2879	0.2938	0.3888	0.2028	0.533	0.2287	0.2637	0.3876	0.503	**0.3095**	**0.3399**
30	0.2879	0.2938	0.3888	0.2028	0.533	0.2287	0.2637	0.3876	0.503	0.3095	0.3399
35	0.2879	0.2938	0.3888	0.2028	0.533	0.2607	0.2637	0.4476	0.5185	0.3095	0.3506
40	0.2879	0.2938	0.3888	0.2028	0.533	0.2607	0.2637	0.4476	0.5185	0.3095	0.3506
45	0.2879	0.2938	0.3888	0.2028	0.533	0.2607	0.2637	0.4476	0.5185	0.3095	0.3506
50	0.2879	0.2938	0.3888	0.2028	0.533	0.2607	0.2637	0.4476	0.5185	0.3095	0.3506

- ***Observation 3:*** After reaching the minimum value 0.3399 percent at safety level 25, the average EER achieved by the verification system C-RI-LI monotonically increases with an increase in the safety level. The most important thing to notice is, though the EER achieved by the verification system C-RI-LI increases, it does not cross the EER achieved by the weighted sum fusion of C, RI, and LI. For example, at safety level 50, the average EER achieved by verification system C-RI-LI reaches 0.3506, which is 0.03 percent less than the average EER achieved by the weighted sum fusion of C, RI, and LI (which is 0.3507).

Note that the behavior of our proposed fusion framework that we observe in Figure 5.8 (with verification system C-RI-LI) is the same as the behavior we observed in Figure 5.7 (with verification system RI-G-LI). Therefore, we can conclude that this is the typical behavior of our proposed fusion framework. The reasons for this behavior are explained in detail in Section 5.2.3, Page 77.

Performance Improvement: Table 5.10 shows the percentage of ***reduction*** in the average EER achieved with the verification system C-RI-LI (given in Table 5.9, column "Average") in comparison to the average EER achieved with the weighted sum fusion of C, RI, and LI (given in Table 5.8, column "Average"). We see that the verification system C-RI-LI achieved a maximum of 3.08 percent reduction in the average EER over the weighted sum fusion of C, RI, and LI. Thus, our proposed fusion framework shows considerable promise in improving the performance of multi-biometric verification systems. In Table 5.10, the percentage reductions in the average EER at safety levels 0 through 20 are negative because the average EERs obtained by the verification system C-RI-LI at safety levels 0 through 20 (given in Table 5.9, column "Average") are higher than the average EER obtained by the weighted sum fusion of C, RI, and LI (given in Table 5.8, column "Average"). We explained this fact in observation 1 in Section 5.2.3, Page 77.

Table 5.10: Percentage of ***reduction*** in the average EER achieved with the verification system C-RI-LI (given in Table 5.9, column "Average") in comparison to the average EER achieved with the weighted sum fusion of C, RI, and LI (given in Table 5.8, column "Average"). The safety level (p) is varied between 0 and 50.

Safety Level (p)	**% Reduction in Average EER**
0	-108.8395
5	-97.0345
10	-30.1967
15	-8.982
20	-11.0636
25	**3.0796**
30	3.0796
35	0.0285
40	0.0285
45	0.0285
50	0.0285

Table 5.11 shows the percentage of reduction in the average EER achieved with the verification system C-RI-LI in comparison to the average EER achieved with the best performing individual verifier C (given in Table 5.2, column "Average"). Specifically, we compared the minimum of the average EERs obtained by the verification system C-RI-LI (0.3399 percent) with the average EER obtained by the best individual verifier C (4.2579 percent). We observe that the proposed fusion framework (verification system C-RI-LI) reduces the equal error rate significantly (92.0172 percent).

Table 5.11: Percentage of reduction in the average EER achieved with verification system C-RI-LI in comparison to the average EER achieved with the best performing individual verifier C (given in Table 5.2, column "Average").

Verification System	% Reduction in Average EER
C-RI-LI	92.0172

Convenience to Genuine Users: Table 5.12 shows the percentage of genuine users getting verification decisions at different stages of the four-stage multi-biometric verification system C-RI-LI when safety level (p) is varied between 0 and 50. In other words, Table 5.12 shows how much convenience our proposed framework (with verification system C-RI-LI) provides to the genuine users. Let us give an example. With verification system C-RI-LI, we achieved the minimum EER 0.3399 percent at a safety level of 25, which is 3.08 percent less than the EER obtained with the weighted sum fusion of C, RI, and LI (0.3507 percent). Now we will see how much convenience we achieve with the verification system C-RI-LI at safety level 25.

In Table 5.12, we see that at safety level 25, 25.89 percent of the genuine users received verification decision by using only one biometric trait (C), *i.e.*, 25.89 percent of the genuine users did not need to submit a second or third biometric trait. Also, 57.29 percent of the genuine users received verification decision by using only two biometric traits (C and RI), *i.e.*, (25.89 + 57.29) or 83.18 percent of the genuine users received verification decisions without submitting the third biometric trait (LI). In contrast, in the case of the weighted sum fusion, a user is bound to submit all of the three biometric traits (C, RI, and LI). Thus, the proposed fusion framework gives the minimum EER, and at the same time, it provides a considerable amount of convenience to the genuine users.

Table 5.12: Percentage of genuine users getting verification decisions at different stages of the four-stage multi-biometric verification system C-RI-LI, when safety level (p) is varied between 0 and 50.

Safety Level	**1st Stage**	**2nd Stage**	**3rd Stage**	**4th Stage**
(p)	**(C)**	**(RI)**	**(LI)**	**(Parallel Unit)**
0	47.64	43.64	3.29	5.43
5	43.33	46.59	3.49	6.59
10	38.18	50.12	4.03	7.67
15	33.8	52.64	4.81	8.76
20	29.46	55.23	4.84	10.47
25	**25.89**	**57.29**	**5.04**	**11.78**
30	21.51	59.34	5.85	13.29
35	18.99	59.61	6.67	14.73
40	15.19	60.43	7.52	16.86
45	11.86	61.43	8.33	18.37
50	9.42	61.09	9.11	20.39

CHAPTER 6

PROPOSED FRAMEWORK FOR CONTINUOUS KEYSTROKE VERIFICATION WITH WEAK TEMPLATES

In this chapter, we propose a framework comprised of *impostor score based normalization*, *impostor score based rejection*, and *fusion* to lower the EERs of continuous keystroke verification with weak templates. We introduce a *new formulation* to incorporate the reject option in verification with weak templates and develop a new *impostor score based* rejection method, called the Order Statistic (OS) rejection method. Furthermore, we adapt: 1) the Otsu threshold selection method [40], and 2) the Gaussian assumption of scores to our rejection formulation and study how they perform as *impostor score based* rejection methods. We conduct experiments on a large keystroke database of ***1100*** users.

The rest of the chapter is organized as follows. In Section 6.1, we describe the proposed framework. In Section 6.2, we formulate the impostor score based rejection and present three rejection methods. In Section 6.3, we discuss data collection and experiment design. In Section 6.4, we discuss the experimental results.

6.1 Proposed Framework

We propose a framework for continuous keystroke verification with weak templates. Figure 6.1 shows the proposed framework. It consists of four modules–1) score generation by individual verifiers, 2) impostor score based normalization, 3) rejection, and 4) fusion.

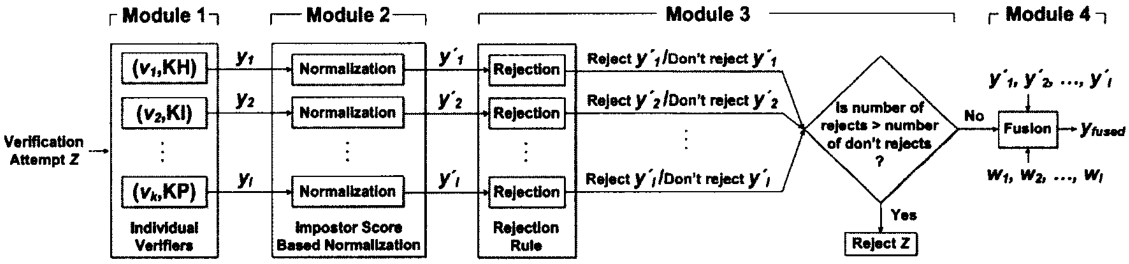


Figure 6.1: Schematic diagram of normalization, rejection, and fusion based continuous keystroke verification framework.

Let $\{v_1, \cdots, v_k\}$ denote k verifiers. Let KH, KI, and KP denote templates containing key hold, key interval, and key press latencies, respectively. Let $\{(v_1$, KH), $(v_1$, KI), $(v_1$, KP), $(v_2$, KH), $\cdots$, $(v_k$, KP)$\}$ be a set of verifier-template pairs. A verifier-template pair, say $(v_1$, KP), means that verifier v_1 uses the template containing key press latencies to generate verification scores. We paired each verifier with a template containing either key hold or key interval or key press latencies because the keystroke verifiers used in this paper (*i.e.*, "R", "A", and "S") were not designed to work with templates containing multiple *types* of latencies. For example, the verifiers are not designed to work with a template containing both key hold and key interval latencies.

Module 1 (see Figure 6.1) consists of a set of verifier-template pairs and generates raw scores for a verification attempt. Let Z be a verification attempt and let y_1, y_2, $\cdots$, y_l be the raw scores of Z generated by (v_1, KH), (v_1, KI), $\cdots$, (v_k, KP), respectively.

Module 2 performs impostor score based normalization [68]. In impostor score based normalization, each user U has a set of (training) impostor scores with each verifier-template pair (*i.e.*, U has a set of impostor scores with (v_1, KH), another set of scores with (v_1, KI), and so on). These impostor scores are generated by matching U's template with a set of impostor attempts, which are different from impostor attempts used in testing. Using each set of impostor scores, we estimate the impostor score density. We used z-score normalization in which each raw score, say y_l, is transformed as $y'_l = (y_l - \mu_l)/\sigma_l$, where μ_l and σ_l are the mean and standard deviation of the corresponding impostor score density.

Module 3 performs rejection [29], [44]. This module checks whether the verification attempt Z is good enough to be classified as genuine or an impostor. Z is rejected if the verifier cannot determine whether Z is genuine or an impostor. A rejection rule is applied to each normalized score y'_l to determine whether y'_l should be rejected or not. If more than half of $y'_1, y'_2, \cdots, y'_l$ get rejected by the rejection rule, then verification attempt Z is rejected; otherwise, $y'_1, y'_2, \cdots, y'_l$ are fused to generate score y_{fused}. The rejection rules are described in Section 6.2.

Module 4 performs fusion [1], [69]. The goal of fusion is to use $y'_1, y'_2, \cdots, y'_l$ to obtain a fused score y_{fused}. For fusion, we implemented the weighted sum rule, *i.e.*, $y_{fused} = w_1 y'_1 + \cdots + w_l y'_l$, where y' is a normalized score and the weights are constrained as $w_1 + \cdots + w_l = 1$. In Modules 2 and 4, we chose z-score normalization and weighted sum fusion because they were easy to implement and have been shown to perform well in score-level fusion studies [1].

6.2 Rejection Methods

The purpose of adding a rejection rule to verification with weak templates is to reduce EERs by rejecting as few genuine verification attempts as possible. Ambiguity and distance rejection *cannot* be directly applied to verification with weak templates for the following reasons: 1) ambiguity-reject assumes that all classes are known and the posterior probabilities of classes can be estimated well. However, in verification with weak templates, we can estimate the density of one class (*i.e.,* impostor), while the density of the other class (*i.e.*, genuine) is *unknown*; and 2) distance-reject rejects a pattern based on how far it is from the samples or prototypes of known classes. However, in verification with weak templates, scores of genuine verification attempts are (ideally) expected to be far from the impostor scores. So, applying distant-reject directly can result in erroneously rejecting many genuine attempts.

Our Formulation: To incorporate the reject option in verification with weak templates, we modify the distance-reject rule by using two thresholds to identify a *reject region.* In our formulation, the reject region is the region in which unknown genuine scores overlap with known impostor scores. Let $x_1, \cdots, x_n$ represent n impostor scores generated by a verifier-template pair (v, f) for user U. Let X be the random variable representing these impostor scores. Let f_X represent the density function of X. For expositional convenience, we assume that 1) the verifier outputs a *dissimilarity* score (*i.e.*, genuine scores typically have smaller values than impostor scores), and 2) all scores lie in $[0, 1]$. However, our formulation can be straightforwardly extended to any type of verifier (that outputs *similarity* scores and scores outside $[0, 1]$ range).

Figure 6.2 shows the proposed formulation. We define reject region R as the impostor score region bounded by lower threshold LT and upper threshold UT ($LT, UT \in [0, 1]$). Reject region R has two parameters–1) *position* as specified by the closed interval $[LT, UT]$, and 2) *size*, defined as $\Delta = UT - LT$, $\Delta \in [0, 1]$. The cumulative probability of R is $C = \int_{LT}^{UT} f_X dX$. The value of C is maximum (*i.e.*, 1) when R is $[0, 1]$ and minimum (*i.e.*, 0) when $LT = UT$. In our formulation, R represents a window in which genuine and impostor scores overlap and the idea is to reduce EER by rejecting the verification scores which fall in R. EER values change as R changes its position and size. Therefore, by using our formulation, we can control the EERs by changing the position and size of R.

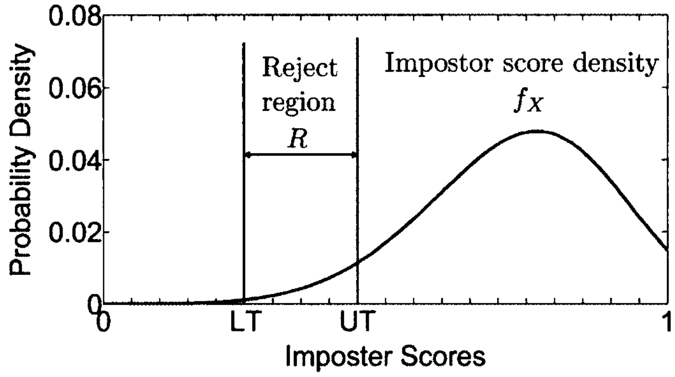


Figure 6.2: Formulation of impostor score based rejection.

Our formulation poses the challenge: how to choose thresholds LT and UT of R under the constraint that genuine scores are *not* given? We introduce three methods (described below) to address the challenge.

6.2.1 Order Statistic Rejection Method

Recall $x_1, x_2, \cdots, x_n$ are n independent impostor scores of user U, X is a random variable representing the impostor scores, and f_X is the density function of X. Let F_X denote the cumulative distribution function of X. Let $x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$ represent the *order statistics* of $x_1, x_2, \cdots, x_n$, *i.e.*, $x_{(r)}$ denotes the impostor score at rank r when $x_1, x_2, \cdots, x_n$ are arranged in ascending order. Each rank r can be associated with an impostor risk, IR(r), which can be empirically calculated as

$$\mathrm{IR}(r) = \frac{\text{number of impostor scores} \leq x_{(r)}}{n}. \tag{6.1}$$

Equation (6.1) gives the impostor risk of r, estimated from a single realization of n impostor scores $x_1, \cdots, x_n$. Estimating the impostor risk of r with (6.1) is unreliable because it is based on only one instance (*i.e.*, a snapshot) of n impostor scores. A more reliable approach to find IR(r) is to calculate the expected impostor risk of rank r when n scores are independently and identically drawn from f_X.

Let $K = P(X \leq x_{(r)}) = F_X(x_{(r)})$ be a cumulative probability value, *i.e.*, K is the proportion of impostor scores less than or equal to $x_{(r)}$. Assuming that F_X is the *true* cumulative distribution function of X (*i.e.*, the functional form and the parameters of F_X are somehow known apriori and not estimated), K gives the impostor risk of rank r. The following formulas are well known (see [70], [71]):

$$P(K) = r\binom{n}{r}(1-K)^{n-r}(K)^{r-1} \tag{6.2}$$

and

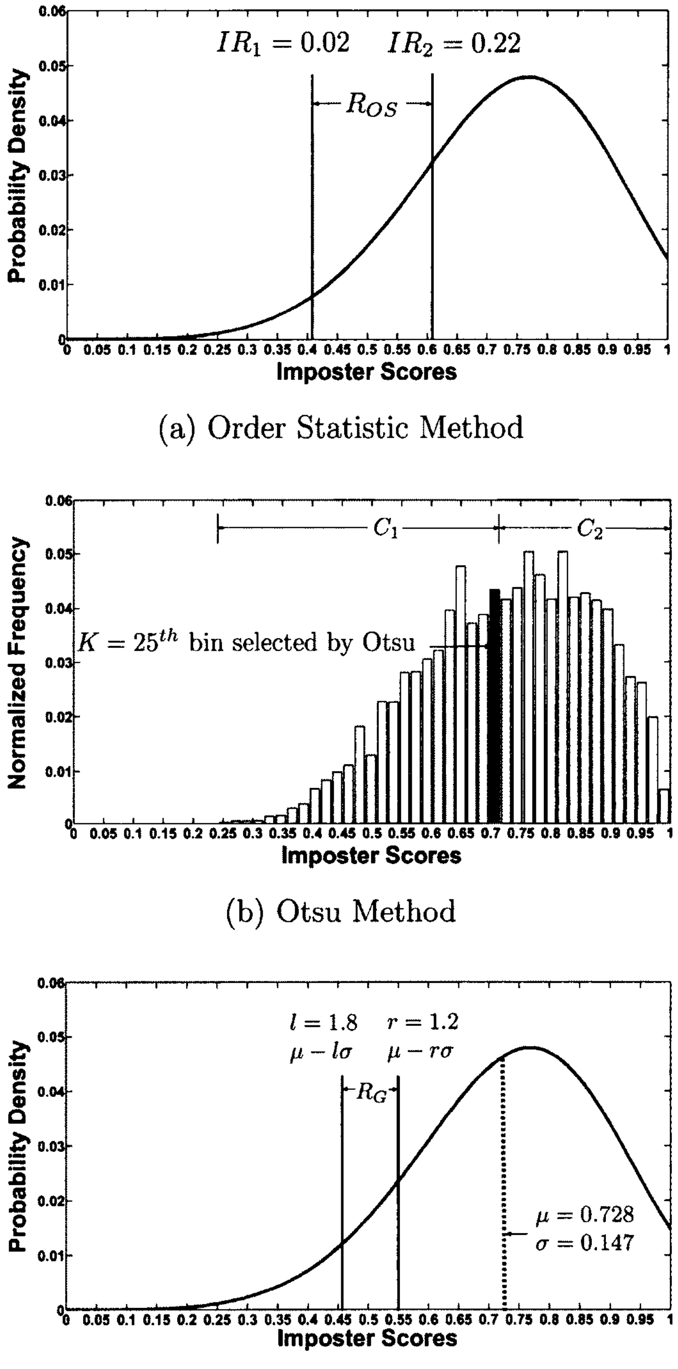
$$E(K) = \int_0^1 KP(K)dK = \frac{r}{n+1}, \tag{6.3}$$

where $P(K)$ is the probability density function of K and $E(K)$ is the expected value of K. Equations (6.2) and (6.3) show that both the density function and the expectation of K are *independent* of the impostor score density f_X. Equation (6.3) further implies that the expected impostor risk of rank r is *dependent* only on the rank r and the number of impostor scores n, regardless of the actual impostor score value $x_{(r)}$ or the cumulative distribution function F_X.

We determine Order Statistic (OS) reject region, R_{OS}, as follows- *STEP 1)* Specify two impostor risks IR_1 and IR_2; *STEP 2)* Using (6.3), find ranks r_1 and r_2 that correspond to IR_1 and IR_2 (*i.e.*, $r_1 = (n+1) \times IR_1$ and $r_2 = (n+1) \times IR_2$); and *STEP 3)* R_{OS} is the region bounded by $x_{(r_1)}$ and $x_{(r_2)}$, where $x_{(r)}$ is the impostor score at rank r when the impostor scores $x_1, x_2, \cdots, x_n$ are arranged in ascending order. Thus, the OS rejection method identifies a reject region using the impostor pass rates associated with ranks, without making any assumptions on the underlying distribution of impostor scores. After R_{OS} is determined, we reject all verification scores falling in R_{OS}. Figure 6.3(a) illustrates the OS rejection method.

6.2.2 Otsu Rejection Method

The Otsu method (developed by Nobuyuki Otsu [40]) is a non-parametric threshold selection method. The input to Otsu is a normalized histogram (*i.e.*, probability distribution) of discretized scores. Otsu iteratively searches for the histogram bin k that partitions the scores into two clusters C_1 (containing scores



(a) Order Statistic Method

(b) Otsu Method

(c) Gaussian Method

Figure 6.3: Impostor score based reject region identification with (a) Order Statistic, (b) Otsu, and (c) Gaussian methods.

in bins $1, \cdots, k$) and C_2 (containing scores in bins $k+1, \cdots, L$), where L is the number of bins in the histogram and k is the bin that maximizes the discriminant criterion: $\frac{\sigma^2_{inter}(k)}{\sigma^2_{inter}(k)+\sigma^2_{intra}(k)}$. Here, $\sigma^2_{inter}(k)$ and $\sigma^2_{intra}(k)$ are inter-cluster and intra-cluster variances of clusters C_1 and C_2 partitioned by the k^{th} bin, given as $\sigma^2_{inter}(k) = \omega_1(k)\omega_2(k)[\mu_2(k) - \mu_1(k)]^2$ and $\sigma^2_{intra}(k) = \omega_1(k)\sigma^2_1(k) + \omega_2(k)\sigma^2_2(k)$, where $\omega_1(k)$ and $\omega_2(k)$ are probabilities of C_1 and C_2, $\mu_1(k)$ and $\mu_2(k)$ are zeroth-order cumulative moments, and $\sigma^2_1(k)$ and $\sigma^2_2(k)$ are first-order cumulative moments of C_1 and C_2, respectively. Figure 6.3(b) illustrates the Otsu method with the 25^{th} bin partitioning the scores into clusters C_1 and C_2. Because k is a bin number in the normalized histogram, the reject region R_{Otsu} can be bounded by the lower and upper limits of k. In Section 6.3.2, we give details on the limits used for R_{Otsu} in our experiments.

Otsu views the reject region identification as a threshold selection problem and does not assume any functional form on the impostor score distribution. However, the drawback with Otsu is it requires a careful selection of histogram bin widths. If the bin is too wide, it over generalizes the score distribution and if too narrow, it overfits the distribution.

6.2.3 Gaussian Rejection Method

In this method, we assume that impostor scores are independent and identically drawn from Gaussian distribution, *i.e.*, $f_X \sim N(\mu, \sigma)$. Because the verifiers we use output *dissimilarity* scores between $[0, 1]$, the genuine scores are expected to be towards the left side of the impostor score distribution. Therefore, the reject region is defined towards the left tail, as shown in Figure 6.3(c). The reject region, R_G, is bounded by

$[\mu - l\sigma, \mu - r\sigma]$, where l is the left coefficient, r is the right coefficient, and $l > r > 0$. The mean (μ) and standard deviation (σ) are the maximum likelihood estimates. In Section 6.3.2, we give details on the l and r values used in our experiments. We implemented the Gaussian rejection method because of its simplicity and use it for comparison.

6.3 Experiments

6.3.1 Keystroke Data Collection

We collected keystroke data at Louisiana Tech University. Data were collected during two periods–1) between April 4 and April 30, 2010 and 2) between October 25 and November 9, 2010. We collected at least 1800 characters of *copy text* and at least 300 characters of *self text* from each participant. Participants typed copy text from the text samples provided by us. Participants had to compose self text by themselves, as if they were writing an email or composing an essay. While typing copy and self texts, participants were allowed to correct typos and use any key on the keyboard.

For training (*i.e.*, building weak templates from samples collected in a single enrollment session), we used keystroke samples collected from 100 participants during April 4-April 30, 2010. We call this dataset D_1. For generating impostor scores to be used in impostor score based normalization and impostor score based rejection, we used samples collected during Oct. 25-Nov. 9, 2010 from 500 participants who were not in D_1. We call this dataset D_2. The test dataset D_3 contains samples collected during Oct. 25-Nov. 9, 2010 from the same 100 participants who were in D_1 (*i.e.*, test samples were collected approximately six months after the training samples).

Keystrokes from 500 new participants (not present in D_1 and D_2) collected during Oct. 25-Nov. 9, 2010 were included in D_3 for generating zero-effort impostor attempts. In summary, a user's weak template, constructed with a keystroke sample in D_1, was matched against his/her verification attempts in D_3 and against impostor attempts from 599 remaining users in D_3.

6.3.2 Design of Experiments

Baseline: We experimented with ten different matching pairs (Ms) and nine verifier-template pairs (verifiers "R", "S", and "A" trained on templates containing key interval (KI), key hold (KH), and key press (KP) latencies). We used dataset D_1 for training and D_3 for testing. Columns KI, KP, and KH in Table 6.1 give the average genuine and impostor verification attempts *per user*, (*i.e.*, total genuine attempts/total users and total impostor attempts/total users), used for obtaining EERs of individual verifiers (Table 6.2). Table 6.1 shows that we had very few genuine verification attempts for high M values like 300, 350, and 500. Recall that D_1 has 100 users, so the total users are 100. In test data, there were on average 2934 keystrokes per user. Each user typed on average 751.33, 879.47, and 1251.31 keystrokes to generate a verification attempt with M values 300, 350 and 500, respectively. So, for high M like 500, 350, and 300, we could only extract between 2 to 4 genuine verification attempts per user. However, we did not have this problem with impostor attempts because, for each user, we had 599 users to generate zero-effort impostor attempts.

Order Statistic Rejection: We tested 250 reject regions. First, we initialized IR_1 to 20 values, from 0.02 to 0.2 (*i.e.*, IR_1 was shifted right towards the

Table 6.1: Average genuine (G) and impostor (I) verification attempts *per user* used in the experiments. Verification attempts in column KI (key interval), KP (key press), and KH (key hold) were used to generate the results in Table 6.2. Verification attempts in column Fusion were used to generate the results in tables 6.3, 6.4 and 6.5.

M	KI		KP		KH		Fusion	
	G	I $\times 10^3$	G	I $\times 10^3$	G	I $\times 10^3$	G	I $\times 10^3$
20	58.9	35.5	58.6	35.2	71.1	42.6	56.0	33.8
30	39.1	23.5	38.8	23.4	47.2	28.3	37.3	22.5
40	29.2	17.6	29.1	17.5	35.4	21.2	27.9	16.8
50	23.3	14.0	23.1	13.9	28.1	16.9	22.2	13.4
60	19.3	11.6	19.2	11.5	23.4	14.0	18.5	11.1
100	11.3	6.9	11.3	6.8	13.8	8.3	10.8	6.5
150	7.4	4.5	7.4	4.4	9.1	5.4	7.1	4.3
300	3.5	2.1	3.5	2.1	4.2	2.5	3.3	2.0
350	2.9	1.7	2.8	1.7	3.7	2.2	2.8	1.7
500	1.9	1.1	1.9	1.1	2.2	1.3	1.8	1.1

impostor score distribution, in increments of 0.02). Next, for each IR_1 value, we set IR_2 as $IR_1 + 0.2$ and shifted IR_2 rightwards in increments of 0.01 until it reached a maximum value of 0.55. For example, when $IR_1 = 0.02$, IR_2 was varied as 0.22, 0.23, 0.24, and so on, until 0.55. This process was repeated again with IR_1 set to 0.04, 0.06, and so on, till 0.2. This way, we created 250 different reject regions.

Otsu Rejection: We tested 256 reject regions. Because Otsu outputs a bin number, we extracted a real value threshold T from the bin by setting T to the upper limit of the bin. We calculated R_{Otsu} as $(T - \delta_1, T + \delta_2)$, where δ_1 and δ_2 take 16 different real values, from 0.01 to 0.16, in increments of 0.01. We recorded the reject rates and corresponding EERs with all 256 (*i.e.*, 16 × 16) combinations of δ_1 and δ_2. We chose to use 40 equal-width bins after performing trial and error experiments.

Table 6.2: EERs of three continuous keystroke verifiers "R", "S", and "A" trained with key interval (KI), key hold (KH), and key press (KP) templates when M is varied between 20 and 500.

M	R			S			A			Avg	SD
	KI	KP	KH	KI	KP	KH	KI	KP	KH		
20	0.235	0.293	0.282	0.239	0.27	0.38	0.33	0.258	0.275	0.285	0.0458
30	0.202	0.26	0.249	0.205	0.239	0.35	0.304	0.235	0.248	0.255	0.0468
40	0.18	0.238	0.226	0.186	0.221	0.331	0.29	0.22	0.237	0.237	0.0476
50	0.166	0.22	0.216	0.17	0.198	0.315	0.281	0.212	0.23	0.223	0.0484
60	0.157	0.211	0.205	0.164	0.194	0.291	0.28	0.208	0.223	0.215	0.0456
100	0.137	0.182	0.18	0.142	0.172	0.284	0.254	0.19	0.208	0.194	0.0483
150	0.122	0.16	0.164	0.123	0.16	0.28	0.238	0.183	0.207	0.182	0.0521
300	0.117	0.145	0.144	0.11	0.14	0.265	0.236	0.178	0.185	0.169	0.0528
350	0.116	0.15	0.14	0.112	0.124	0.258	0.226	0.178	0.19	0.166	0.0511
500	0.105	0.145	0.126	0.111	0.122	0.252	0.235	0.182	0.18	0.162	0.0538

Gaussian Rejection: We tested 120 reject regions. First, we initialized (l, r) to 15 values - (0.2, 0.0), (0.3, 0.1), (0.4, 0.2), and so on, until (1.6, 1.4) (*i.e.*, l and r were increased in increments of 0.1). Next, we expanded each (l, r) seven times with a step size of 0.05. For example, (1.6, 1.4) was expanded as (1.65, 1.35), (1.7, 1.3), and so on, till (1.95, 1.05). This way, we created 120 different reject regions.

Fusion: In fusion, we used five verifier-template pairs: (R, KH), (R, KI), (S, KH), (S, KI), and (A, KP). As key press (KP) latencies are formed by adding key hold (KH) and key interval (KI) latencies [72], KP does not bring new information if KH and KI are already included. Therefore, with verifiers "R" and "S", we paired KH and KI. However, with verifier "A", we used KP because we needed an odd number of verifier-template pairs (for avoiding ties during rejection with majority voting) and "A" was designed primarily for KP and performed better with KP than with KI or KH (see Table 6.2).

Setting Parameters for Rejection Rules: The main goal of our experiments is to comprehensively evaluate the performance of our rejection rules across a broad range of parameters. Therefore, we computed EERs with 250 (IR_1, IR_2), 256 (δ_1, δ_2), and 120 (l, r) parameter values. However, in an actual verification scenario, the parameter values have to be set. Optimal parameter values can be estimated using well-known parameter selection techniques such as cross-validation on a hold-out dataset, bootstrapping [64], and genetic algorithm [65].

In the rejection and fusion experiments, we used databases D_1 for training, D_2 for generating impostor score densities, and D_3 for testing.

6.4 Results and Analysis

6.4.1 Baseline Results

In Table 6.2, we give EERs of "R", "S", and "A" verifiers trained with templates containing key interval (KI), key hold (KH), and key press (KP) latencies. We tested each verifier-template combination with ten different matching pairs (M varied between 20 and 500). The "Avg" and "SD" columns give the means and standard deviations of EERs at each M calculated over the nine verifier-template combinations. Table 6.2 shows that EERs of almost all verifier-template combinations decrease as M increases. This is expected because verifiers get more information for decision making when M increases. We achieved 0.285 average EER with $M = 20$ and 0.162 with $M = 500$ (*i.e.*, 43.16 percent reduction in average EER when M changed from 20 to 500). Though increasing M reduces EER, the user is required to type more text to generate more matching pairs, which ultimately increases the verification delay. In our experiments, each user on average typed between 12.77 seconds (with KH) to 14.81 seconds (with KP) when $M = 20$ and 307.81 seconds (with KH) to 366.78 seconds (with KP) when $M = 500$ (*i.e.*, the verification delay increased by more than 20 times when M changed from 20 to 500). Therefore, a trade-off exists between verification accuracy and verification delay and the challenge is to achieve lower EERs with smaller Ms.

6.4.2 Results without Incorporating Rejection

To observe the effect of the impostor score based normalization and fusion in performance improvement, we did some experiments without incorporating rejection before fusion and compared the resultant EERs to the baseline EERs. Recall that we fused five verifier-template pairs: (R, KH), (R, KI), (S, KH), (S, KI), and (A, KP). We conducted the experiments with 7 different M values (varied between 20 and 150). At each M, we obtained EERs with 21 different weight combinations. The number of genuine and impostor verification attempts per user used in these experiments is given in Table 6.1, "Fusion" column. In Table 6.3, in the "Best" column, we give the lowest EER obtained among 21 weight combinations. In "Avg" and "SD" columns, we give the average EERs and standard deviations obtained with 21 weight combinations. For $M = \{20, 30, 40, 50, 60\}$, the best EERs were achieved with equal weights (0.2, 0.2, 0.2, 0.2, 0.2). For $M = \{100, 150\}$, the best EERs were achieved with (0.266, 0.266, 0.266, 0.1, 0.1) weights.

Table 6.3: EERs with impostor score based normalization and fusion when M is varied between 20 and 150. "% Reduction" column gives the percentage of *reduction* in the average EERs achieved with impostor score based normalization and fusion compared to the individual verifiers (Table 6.2).

M	**Best**	**Avg**	**SD**	**% Reduction**
20	0.124	0.1497	0.0228	47.47
30	0.0999	0.1110	0.0065	56.47
40	0.0864	0.0958	0.0062	59.57
50	0.0769	0.0855	0.0059	61.65
60	0.0693	0.0772	0.0057	64.09
100	0.0575	0.0639	0.0048	67.06
150	0.0454	0.0552	0.0068	69.67

Performance Improvement: Table 6.3, "% Reduction" column, gives the percentage of *reduction* in the average EERs achieved with impostor score based normalization and fusion compared to the individual verifiers (Table 6.2). With normalization and fusion, we achieved 47.47 to 69.67 percent *reduction* in average EERs over the individual verifiers. Thus, the impostor score based normalization and fusion shows considerable promise in improving continuous verification EERs, especially with low Ms (20, 30, 40, 50, and 60), which had less than one minute verification delay in fusion experiments (each user on average typed between 13.42 seconds (when $M = 20$), and 46.09 seconds (when $M = 60$) to generate a verification attempt). We excluded $M = \{300, 350, 500\}$ in the fusion experiments because a user on average typed for 227.64 seconds (when $M = 300$), 267.06 seconds (when $M = 350$), and 380.75 seconds (when $M = 500$) to generate a verification attempt. Verification delays with $M = \{300, 350, 500\}$ are too high (and therefore impractical) for continuous verification applications.

6.4.3 Results with Incorporating Rejection

In this experiment we incorporate rejection before fusion. We fused five verifier-template pairs: (R, KH), (R, KI), (S, KH), (S, KI), and (A, KP)), with equal weight (0.2) assigned to each pair. We used equal weights because we achieved the best results with equal weights in the baseline fusion experiments. We present the performance of three rejection methods: Order Statistic (OS), Otsu, and Gaussian. We measure performance in terms of equal error rate obtained when x percent of genuine verification attempts are rejected (*i.e.*, EER at x percent genuine rejection). The parameters used for obtaining EERs in this section are discussed in Section 6.3.2. As the size and position of the reject regions change, different percentages of genuine and impostor verification attempts are rejected. While rejection of an impostor attempt incurs no cost (and may even be desirable in multi-factor authentication applications where a rejected attempt can be used to trigger a different authentication factor), rejecting genuine verification attempts causes user inconvenience and incurs costs in terms of re-authentication effort. Therefore, we evaluated the rejection methods using the *percentage of genuine rejection* versus *EER* curves that plot the percentage of genuine attempts erroneously rejected on the x-axis and the corresponding EERs on the y-axis.

Which Rejection Method Performed Better? In Figure 6.4, we plot the average EERs achieved with the rejection methods at various percentages of genuine rejections (*i.e.*, average EERs were calculated when the percentage of genuine rejections were between 0 and 2 percent, 2 and 4 percent, and so on, until 28 and 30 percent). Error bars indicating standard deviations are shown only for OS to avoid cluttering the plots. Plots in Figure 6.4 clearly show that OS *outperforms* Otsu and Gaussian *i.e.* OS achieved lower average EERs compared to Otsu and Gaussian (the percentage of genuine rejections versus EER curves of OS are lower than Gaussian and Otsu) for all Ms.

Performance Improvement over Individual Verifiers: Table 6.4 shows the percentage of *reduction* in the average EERs achieved with the OS rejection method in comparison to the average EERs of individual verifiers (Table 6.2), when the percentage of genuine rejections are between 4 and 6, 8 and 10, 14 and 16, 20 and 22, and 28 and 30. In Table 6.4, we highlight the reduction in average EERs achieved with low Ms (20, 30, 40, 50, and 60), which had less than one minute average verification delay. The actual average EER values for all Ms are given in Figure 6.4. Depending on the percentage of genuine attempts rejected, the OS rejection method

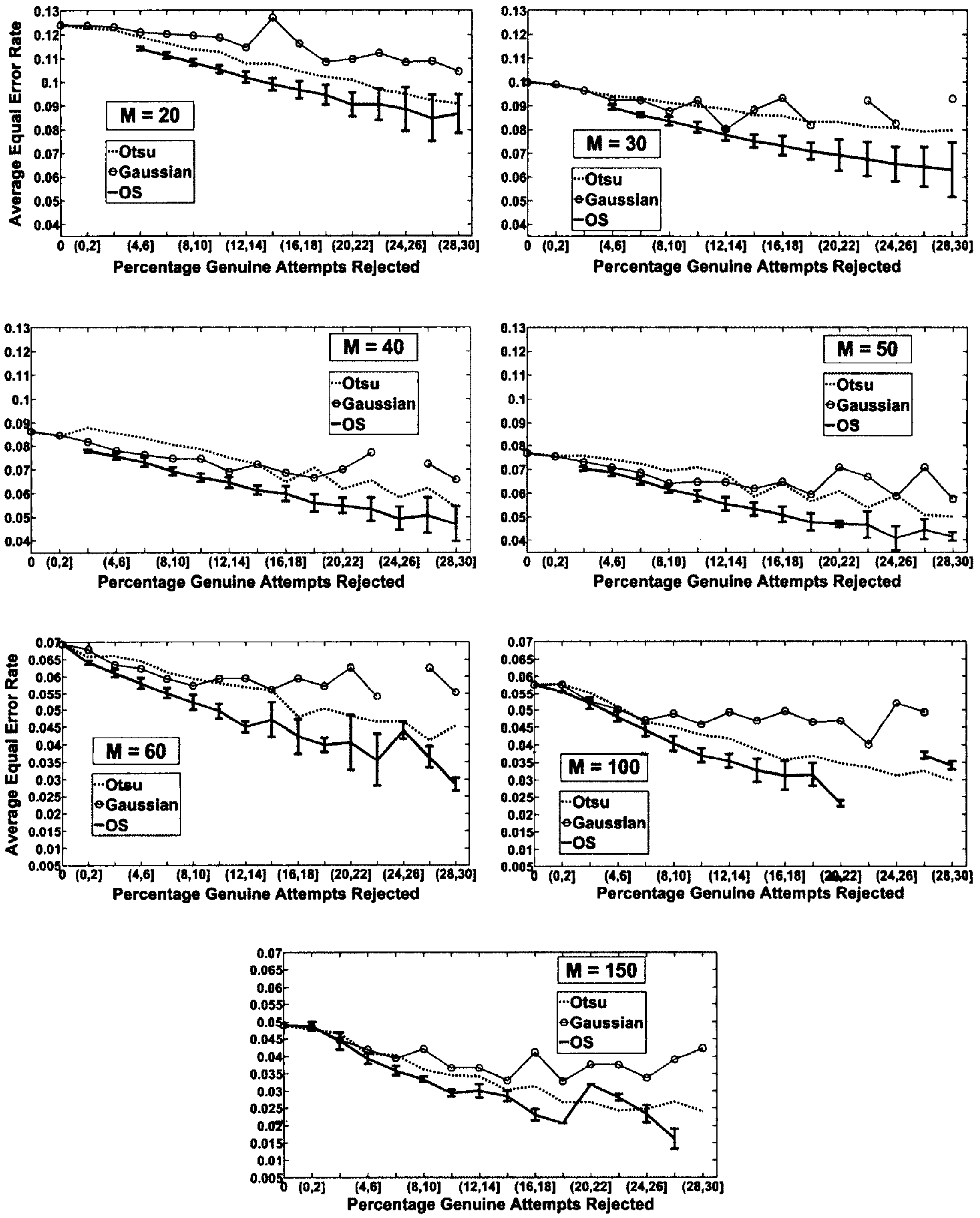


Figure 6.4: Percentage of genuine rejections versus EERs of OS, Otsu, and Gaussian methods with different M values.

achieved 59.97 to 86.74 percent *reduction* in average EERs compared to the individual verifiers.

Table 6.4: Percentage of **reduction** in the average EERs achieved with OS rejection method compared to the individual verifiers (in Table 6.2).

M	**% of Genuine Attempts Rejected**				
	(4-6]	**(8-10]**	**(14-16]**	**(20-22]**	**(28-30]**
20	59.97	61.99	65.24	68.23	69.59
30	64.97	67.23	70.52	72.86	75.3
40	68.11	70.79	74.15	76.94	80.17
50	69.19	72.31	76.14	78.99	81.39
60	73.03	75.72	78.07	81.19	86.74

Performance Improvement over Fusion of Verifiers: Table 6.5 shows the percentage of *reduction* in the average EERs achieved with the OS rejection method in comparison to impostor score based normalization and fusion (Table 6.3). Depending on the percentage of genuine attempts rejected, the OS rejection method achieved between 23.79 percent and 63.07 percent *reduction* in average EERs compared to impostor score based normalization and fusion.

Table 6.5: Percentage of **reduction** in the average EERs achieved with OS rejection method compared to the average EERs achieved with impostor score based normalization and fusion (in Table 6.3).

M	**% of Genuine Attempts Rejected**				
	(4-6]	**(8-10]**	**(14-16]**	**(20-22]**	**(28-30]**
20	23.79	27.64	33.82	39.51	42.1
30	19.52	24.71	32.28	37.66	43.25
40	21.11	27.73	36.05	42.95	50.94
50	19.65	27.78	37.77	45.21	51.45
60	24.89	32.38	38.93	47.61	63.07

CHAPTER 7

CONCLUSIONS

In this dissertation, we propose (1) a new rejection method called the symmetric rejection method for multi-stage biometric verification, (2) a new fusion framework for multi-biometric verification, and (3) a new framework to reduce the equal error rates of continuous keystroke verification with weak templates.

Compared to existing rejection methods, the symmetric rejection method has two notable advantages: (1) it enables the system administrator to control the genuine reject rate and (2) it allows us to calculate the reject region directly from scores without the need to estimate the underlying probability density function. Experiments performed on a four-stage biometric verification system demonstrate significant promise of the symmetric rejection method in multi-stage biometric verification. Using the symmetric rejection method, we achieved (1) a minimum false alarm rate of 0.0039, which is 91.58 percent less than the equal error rate of the top performing individual verifier, and (2) a minimum impostor pass rate of 0.0203, which is 56.16 percent less than the equal error rate of the top performing individual verifier. We compared the performance of the symmetric rejection method with two existing rejection methods: (1) SPRT-based method and (2) Marcialis *et al.*'s method. Results show that to achieve the same value of area under the ROC curve (AUC), genuine users require less number of stages with the symmetric rejection method compared to SPRT-based and Marcialis *et al.*'s rejection methods. This indicates that the symmetric rejection method can provide better user convenience than the existing rejection methods.

The core of our proposed fusion framework is the new concept of "confident reject region" which ensures that the confident unit of the framework incurs zero false alarm rate and zero impostor pass rate. As a consequence, the proposed fusion framework achieves better performance than the parallel fusion framework. This advantage of the proposed fusion framework makes it applicable to very high security applications. In addition, the proposed fusion framework provides convenience to genuine users by allowing them to submit less number of biometric traits than the parallel fusion framework. This advantage of the proposed fusion framework makes it applicable to biometric applications that involve a large population of users or a great number of biometric transactions. We evaluated our proposed fusion framework on two multi-biometric verification systems. Experimental results provide a considerable amount of evidence that the proposed fusion framework improves the performance over the parallel fusion framework, and at the same time, provides a significant amount of convenience to the genuine users.

Our proposed framework for continuous keystroke verification with weak templates consists of impostor score based normalization, impostor score based rejection, and fusion. We introduced a new formulation to incorporate reject option in verification with weak templates and developed a new impostor score based rejection method, called Order Statistic (OS) rejection method. We studied two more impostor score based rejection methods–(1) Otsu and (2) Gaussian, and compared their performance with the OS rejection method. We experimented on a large dataset of 1100 users and evaluated the rejection methods across a broad range of parameter values. Results show that (1) all three rejection methods significantly reduce the EERs (*i.e.* impostor score based rejection has considerable impact on reducing EERs in continuous keystroke verification with weak templates), and (2) the OS rejection method outperforms both Otsu and Gaussian in terms of error-reject trade-off.

In the future, we are interested to see how the proposed rejection methods and fusion framework perform in general two class problems, for example, anomaly detection, fraud detection, etc. In addition, we will work on finding new and interesting human behavioral patterns and anomalies to address security issues in cyberspace and human-computer interaction.

APPENDIX A

THE GENERALIZED DERIVATION OF RELATIONSHIP BETWEEN α_G AND THE UPPER BOUND FOR GENUINE REJECT RATE

In this section, we derive the following relationship between α_G and upper bound for genuine reject rate: *when α_G is equal to $\rho\lambda_G$, where ρ is a rational number such that $0 < \rho \leq 1$, the upper bound for the genuine reject rate is ρK.*

We use Figure A.1 to explain the derivation. In Figure A.1, let B be the EER-threshold and E_1E_2 be the confusion region. λ_G is the proportion of genuine scores in BE_2 and λ_I is the proportion of impostor scores in E_1B. Let $\rho = \frac{q}{m}$, where q and m are positive integers such that $q \leq m$. We divide BE_2 into m parts such that the proportion of genuine scores in each part is $\frac{\lambda_G}{m}$. Similarly, we divide E_1B into m parts such that the proportion of impostor scores in each part is $\frac{\lambda_I}{m}$. Let b_{G_1}, b_{G_2}, $\cdots$, b_{G_m} be the proportions of genuine scores in m parts of E_1B. Because $f_G(x)$ is monotonically decreasing and $f_I(y)$ is monotonically increasing inside the confusion region, b_{G_1}, b_{G_2}, $\cdots$, b_{G_m} are related as: $b_{G_1} \leq b_{G_2} \leq \cdots \leq b_{G_m}$.

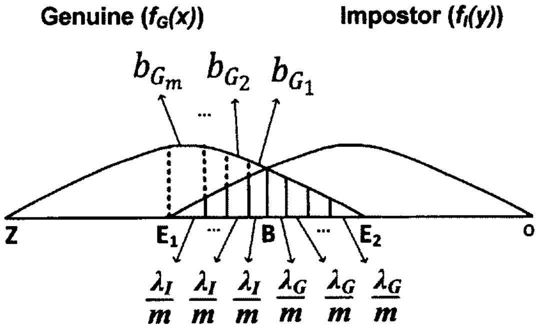


Figure A.1: Dividing E_1B and BE_2 into m parts.

Using the notation in Figure A.1, we can present the proportion of genuine scores in the confusion region E_1E_2, K, as follows:

$$K = b_{G_1} + b_{G_2} + \cdots + b_{G_m} + \lambda_G. \tag{1.1}$$

Now we will explain what happens when $\alpha_G = \rho\lambda_G$. Let AC be a symmetric reject region with $\alpha_G = \rho\lambda_G = \frac{q}{m}\lambda_G = q\frac{\lambda_G}{m}$. Note that we present α_G by q times $\frac{\lambda_G}{m}$. When α_G is equal to q times $\frac{\lambda_G}{m}$, following the symmetric rejection rule, α_I is equal to q times $\frac{\lambda_I}{m}$. As a result, the proportion of genuine scores in AB is equal to $b_{G_1} + b_{G_2} + \cdots + b_{G_q}$. Hence, using Figure A.1, the genuine reject rate is

$$\begin{aligned}
GRR &= \text{Proportion of genuine scores in } AC \\
&= \text{Proportion of genuine scores in } AB + \text{Proportion of genuine scores in } BC \\
&= b_{G_1} + b_{G_2} + \cdots + b_{G_q} + q\frac{\lambda_G}{m} \\
&= \frac{q}{m}\{\frac{m}{q}(b_{G_1} + b_{G_2} + \cdots + b_{G_q}) + \lambda_G\} \\
&= \rho\{\frac{m}{q}(b_{G_1} + b_{G_2} + \cdots + b_{G_q}) + \lambda_G\}. \qquad (1.2)
\end{aligned}$$

Now we will prove that $\frac{m}{q}(b_{G_1} + b_{G_2} + \cdots + b_{G_q}) \leq b_{G_1} + b_{G_2} + \cdots + b_{G_m}$.

If $q = m$, both sides of the above statement become equal. Hence, the statement is true for $q = m$. Below, we prove by contradiction that the statement is true for $q < m$.

For contradiction, we assume that $\frac{m}{q}(b_{G_1} + b_{G_2} + \cdots + b_{G_q}) > b_{G_1} + b_{G_2} + \cdots + b_{G_m}$. This implies that

$$m(b_{G_1} + b_{G_2} + \cdots + b_{G_q}) > q(b_{G_1} + b_{G_2} + \cdots + b_{G_m}).$$

After algebraic manipulation, we get

$$(m - q)(b_{G_1} + b_{G_2} + \cdots + b_{G_q}) > q(b_{G_{q+1}} + b_{G_{q+2}} + \cdots + b_{G_m}).$$

Note that $b_{G_{q+1}} + b_{G_{q+2}} + \cdots + b_{G_m}$ consists of $m - q$ terms. Because $b_{G_1} \leq b_{G_2} \leq \cdots \leq b_{G_m}$, it follows that $b_{G_{q+1}} + b_{G_{q+2}} + \cdots + b_{G_m} > (m-q)b_{G_{q+1}}$. Therefore,

$$(m-q)(b_{G_1} + b_{G_2} + \cdots + b_{G_q}) > q(m-q)b_{G_{q+1}}.$$

Because $q < m$, it follows that $m - q \neq 0$. Hence, we can divide both sides by $m - q$. Dividing by $m - q$, we get $b_{G_1} + b_{G_2} + \cdots + b_{G_q} > qb_{G_{q+1}}$. However, this is impossible because $b_{G_1} \leq b_{G_2} \leq \cdots \leq b_{G_m}$. Hence, the statement $\frac{m}{q}(b_{G_1} + b_{G_2} + \cdots + b_{G_q}) \leq b_{G_1} + b_{G_2} + \cdots + b_{G_m}$ is true. Therefore, we can rewrite (1.2) as follows:

$$GRR \leq \rho(b_{G_1} + b_{G_2} + \cdots + b_{G_m} + \lambda_G). \tag{1.3}$$

Or alternatively, $GRR \leq \rho K$ because $K = b_{G_1} + b_{G_2} + \cdots + b_{G_m} + \lambda_G$. That is, the upper bound for the genuine reject rate is ρK.

APPENDIX B

PROOF FOR LEMMA 4.1

Below, we show that $FAR_i^r < EER_i$. The proof of $IPR_i^r < EER_i$ is similar. First, we introduce two notations. Let PQ be any region in the scoreline $[Z, O]$. Then $P_{G,PQ}$ refers to the proportion of genuine scores in PQ, which is calculated by (the number of genuine scores in PQ)/(total number of genuine scores). Similarly, $P_{I,PQ}$ refers to the proportion of impostor scores in PQ, which is calculated by (the number of impostor scores in PQ)/(total number of impostor scores).

We use Figure 4.1 to explain the proof. In Figure 4.1, let genuine scores and impostor scores originate from verifier v_i, AC be the symmetric reject region, E_1E_2 be the confusion region, and B be the threshold where EER_i occurs.

Using (2.2), the false alarm rate obtained with the symmetric rejection method is

$$\begin{aligned} FAR_i^r &= \frac{\#\text{ of genuine scores in } CE_2}{\text{Total } \#\text{ of genuine scores} - \#\text{ of genuine scores in } AC} \\ &= \frac{\text{Proportion of genuine scores in } CE_2}{1 - \text{Proportion of genuine scores in } AC} \\ &= \frac{P_{G,CE_2}}{1 - P_{G,AC}} \\ &= \frac{P_{G,CE_2}}{1 - P_{G,AB} - P_{G,BC}}. \end{aligned} \tag{2.1}$$

Let $\dfrac{P_{G,BE_2}}{P_{G,BC}} = \mu.$ (2.2)

Because C lies inside $(B, E_2]$, it follows that $\mu \geq 1$. From (2.2), we get $P_{G,BC} = P_{G,BE_2}/\mu$. Hence, we can present the numerator on the right side of (2.1) as follows:

$$P_{G,CE_2} = P_{G,BE_2} - P_{G,BC} = P_{G,BE_2} - \frac{P_{G,BE_2}}{\mu} = \frac{(\mu - 1)P_{G,BE_2}}{\mu}.$$

Therefore, we can rewrite (2.1) as:

$$FAR_i^r = \frac{(\mu - 1)P_{G,BE_2}}{\mu - P_{G,BE_2} - \mu P_{G,AB}}. \tag{2.3}$$

Because EER_i occurs at threshold B,

$$EER_i = P_{I,E_1B} = P_{G,BE_2}. \tag{2.4}$$

We will prove that $FAR_i^r < EER_i$. For contradiction, we assume that $FAR_i^r \geq EER_i$. This implies that

$$\frac{(\mu - 1)P_{G,BE_2}}{\mu - P_{G,BE_2} - \mu P_{G,AB}} \geq P_{G,BE_2}.$$

After algebraic manipulation, we get

$$\mu - 1 \geq \mu - P_{G,BE_2} - \mu P_{G,AB}$$

or alternatively,

$$1 - P_{G,BE_2} \leq \mu P_{G,AB}. \tag{2.5}$$

Because ZE_2 contains all genuine scores, $P_{G,ZE_2} = 1$. Hence, $1 - P_{G,BE_2} = P_{G,ZE_2} - P_{G,BE_2} = P_{G,ZB}$. Therefore, we can rewrite (2.5) as follows:

$$P_{G,ZB} \leq \mu P_{G,AB}. \tag{2.6}$$

Because $f_G(x)$ is monotonically decreasing and $f_I(y)$ is monotonically increasing inside E_1B, the following statement is true:

$$\frac{P_{G,E_1B}}{P_{G,AB}} \geq \frac{P_{I,E_1B}}{P_{I,AB}}. \tag{2.7}$$

From (2.4), $P_{I,E_1B} = P_{G,BE_2}$ and using the symmetric rejection rule, $P_{I,AB} = P_{G,BC}$. Then,

$$\frac{P_{I,E_1B}}{P_{I,AB}} = \frac{P_{G,BE_2}}{P_{G,BC}} = \mu.$$

Hence, we can rewrite (2.7) as follows:

$$\frac{P_{G,E_1B}}{P_{G,AB}} \geq \mu$$

or alternatively,

$$P_{G,E_1B} \geq \mu P_{G,AB}.$$

Because E_1B is a part of ZB (see Figure 4.1), $P_{G,ZB} > P_{G,E_1B}$. Because $P_{G,ZB} > P_{G,E_1B}$ and $P_{G,E_1B} \geq \mu P_{G,AB}$, it follows that $P_{G,ZB} > \mu P_{G,AB}$. However, this contradicts (2.6). Therefore, we conclude that $FAR_i^r < EER_i$.

APPENDIX C

PROOF THAT THE IMPOSTOR PASS RATE OBTAINED BY OUR PROPOSED FUSION FRAMEWORK IS LESS THAN OR EQUAL TO THE IMPOSTOR PASS RATE OBTAINED BY THE PARALLEL FUSION FRAMEWORK

For simplicity, we prove the statement using a specific number of impostors (the proof is similar for any number of impostors). We use Figure C.1 and Figure C.2 to explain the proof.

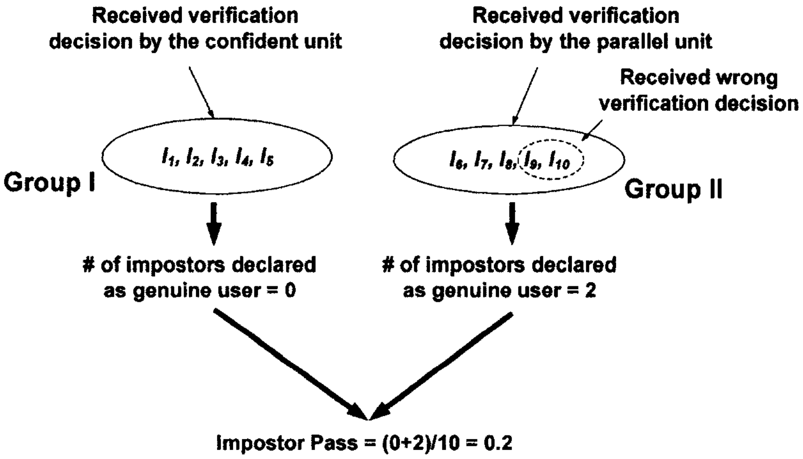

Received verification decision by the confident unit
Received verification decision by the parallel unit
Received wrong verification decision
Group I
I_1, I_2, I_3, I_4, I_5
$I_6, I_7, I_8, I_9, I_{10}$
Group II
# of impostors declared as genuine user = 0
# of impostors declared as genuine user = 2
Impostor Pass = (0+2)/10 = 0.2

Figure C.1: Calculation of impostor pass rate obtained by the proposed fusion framework.

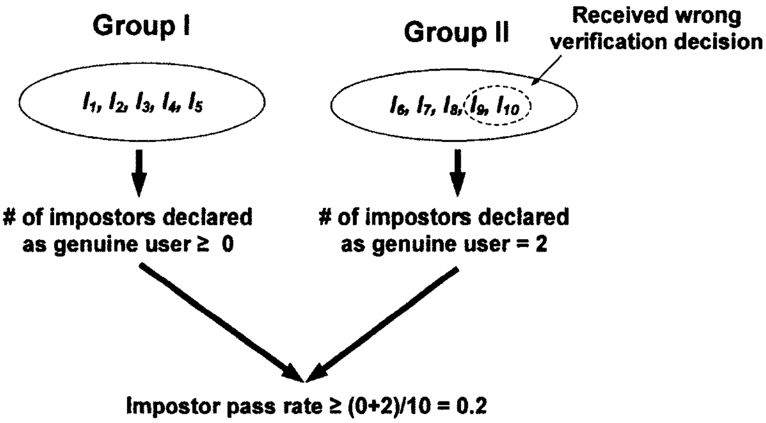

Group I
Group II
Received wrong verification decision
I_1, I_2, I_3, I_4, I_5
$I_6, I_7, I_8, I_9, I_{10}$
# of impostors declared as genuine user ≥ 0
# of impostors declared as genuine user = 2
Impostor pass rate ≥ (0+2)/10 = 0.2

Figure C.2: Calculation of impostor pass rate obtained by the parallel fusion framework.

Let I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9, and I_{10} be 10 impostors. First, we calculate the impostor pass rate obtained by the proposed fusion framework. We use Figure

C.1 in this regard. When we apply the proposed fusion framework, let impostors I_1, I_2, I_3, I_4, and I_5 receive verification decision by the confident unit and the remaining impostors I_6, I_7, I_8, I_9, and I_{10} receive verification decision by the parallel unit of the framework. We define two groups of impostors on the basis of the verification decisions given by the proposed fusion framework:

- Group I: The impostors who receive verification decision by the confident unit. In Figure C.1, Group I consists of impostors I_1, I_2, I_3, I_4, and I_5.
- Group II: The impostors who receive verification decision by the parallel unit. In Figure C.1, Group II consists of impostors I_6, I_7, I_8, I_9, and I_{10}.

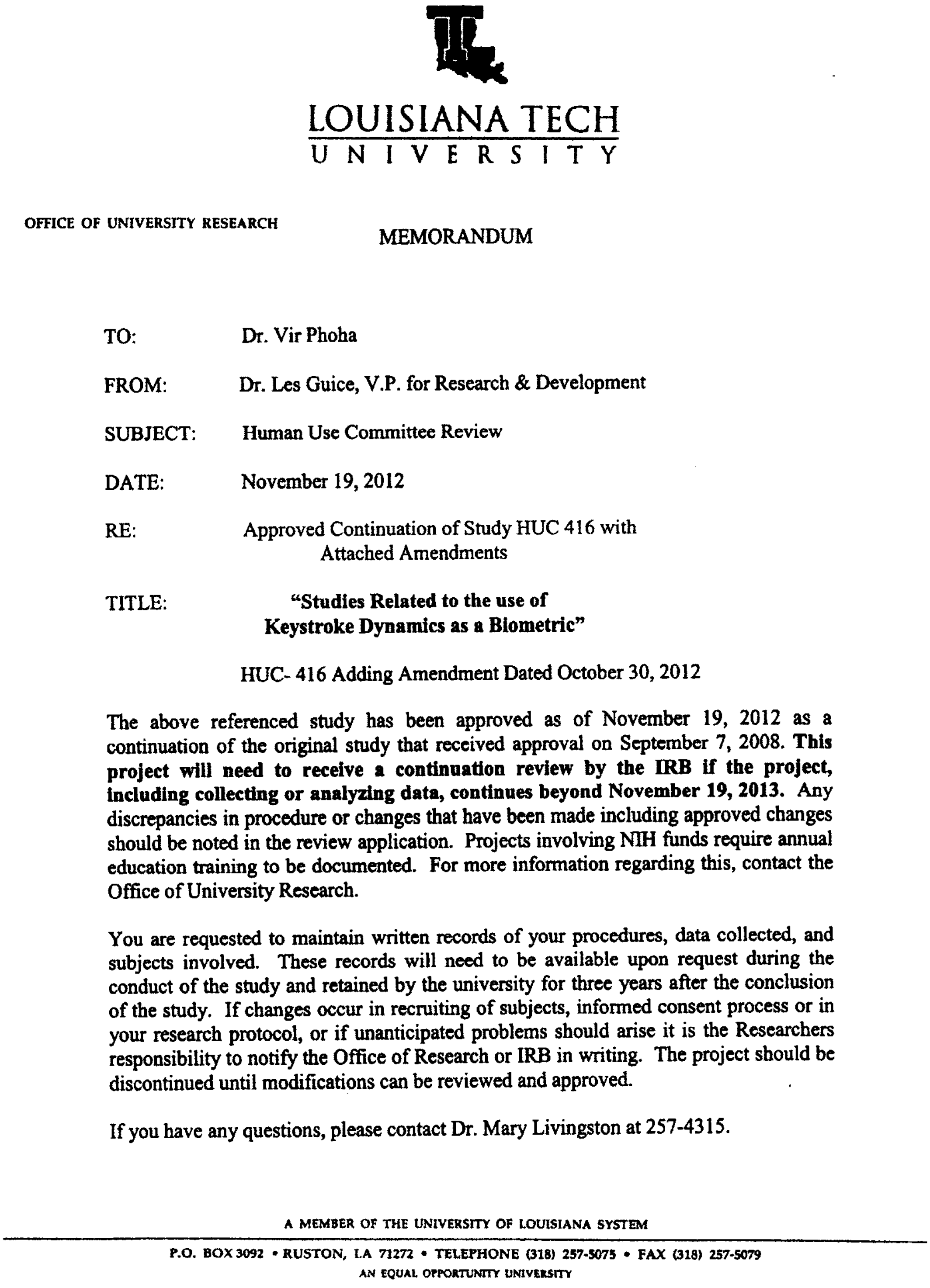
Because the confident unit incurs zero impostor pass rate, the number of impostors in Group I erroneously declared as genuine user is zero. However, because the parallel unit may give some wrong verification decisions, the number of impostors in Group II erroneously declared as genuine user is greater than or equal to zero. Let the parallel unit declare the impostors I_9 and I_{10} as genuine user (for clarity, impostors I_9 and I_{10} are enclosed by a dotted circle in Figure C.1). Hence, the total number of impostors declared as genuine user by the proposed fusion framework is equal to (0+2) or 2. Therefore, the impostor pass rate obtained by the proposed fusion framework is equal to 2/10 or 0.2.

Now, we calculate the impostor pass rate obtained by the parallel fusion framework. We use Figure C.2 in this regard.

In Figure C.1, we apply parallel fusion separately on Group I and Group II. Because parallel fusion may give some wrong verification decisions, when we apply parallel fusion on Group I, the number of impostors in Group I erroneously declared as genuine user is greater than or equal to zero. When we apply parallel fusion on Group II, we achieve exactly the same result as we achieve by applying the parallel unit of the proposed framework on Group II, *i.e.*, the impostors I_9 and I_{10} receive the wrong verification decision. Hence, the total number of impostors declared as genuine user by the parallel fusion framework is greater than or equal to two. Therefore, the impostor pass rate obtained by the parallel fusion framework is greater than or equal to 2/10, *i.e.*, greater than or equal to 0.2.

APPENDIX D

APPROVAL LETTERS FROM THE INSTITUTIONAL REVIEW BOARD OF THE HUMAN USE COMMITTEE



LOUISIANA TECH
UNIVERSITY

OFFICE OF UNIVERSITY RESEARCH

MEMORANDUM

TO: Dr. Vir Phoha

FROM: Dr. Les Guice, V.P. for Research & Development

SUBJECT: Human Use Committee Review

DATE: November 19, 2012

RE: Approved Continuation of Study HUC 416 with
Attached Amendments

TITLE: **"Studies Related to the use of
Keystroke Dynamics as a Biometric"**

HUC- 416 Adding Amendment Dated October 30, 2012

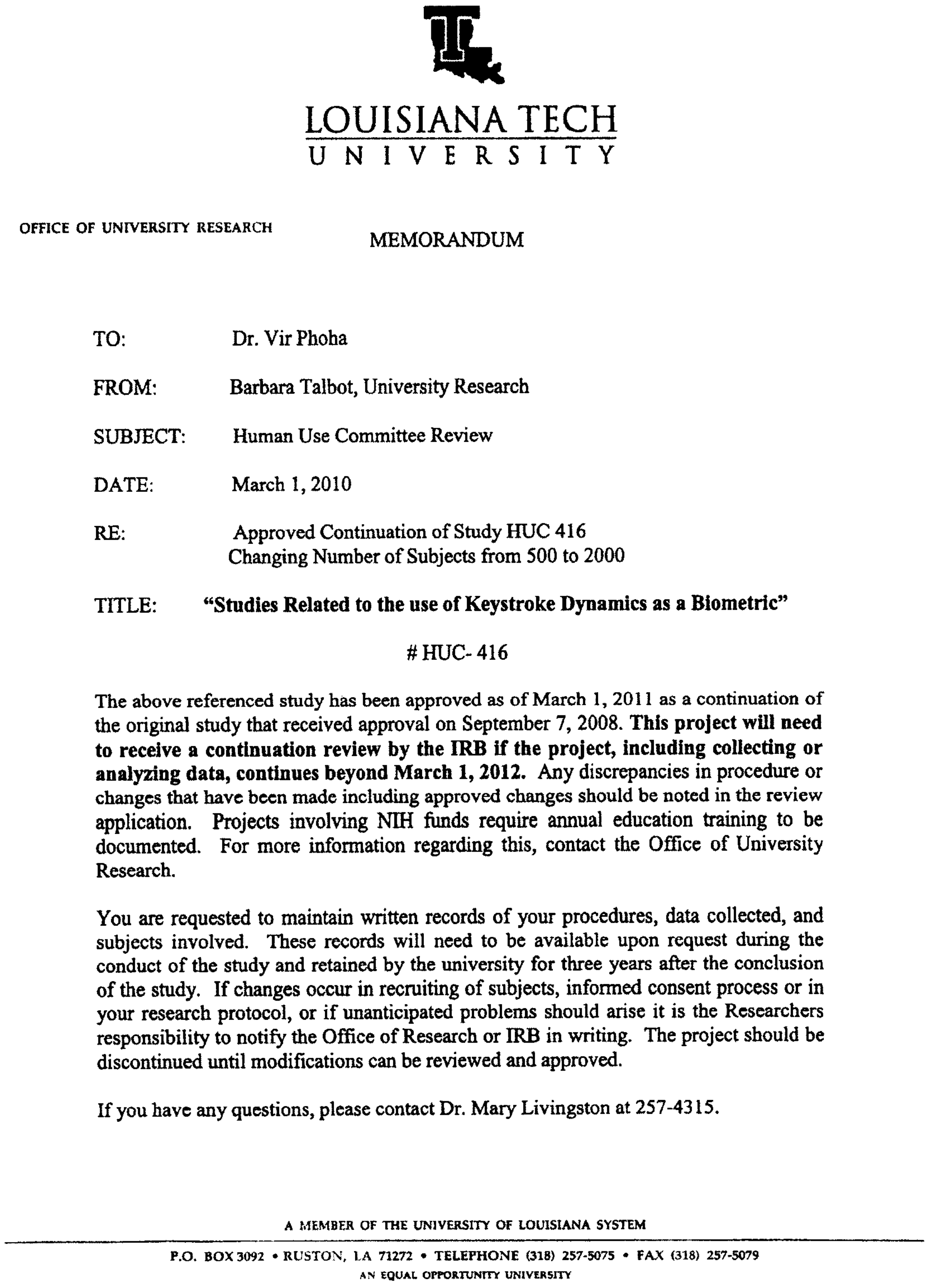
The above referenced study has been approved as of November 19, 2012 as a continuation of the original study that received approval on September 7, 2008. **This project will need to receive a continuation review by the IRB if the project, including collecting or analyzing data, continues beyond November 19, 2013.** Any discrepancies in procedure or changes that have been made including approved changes should be noted in the review application. Projects involving NIH funds require annual education training to be documented. For more information regarding this, contact the Office of University Research.

You are requested to maintain written records of your procedures, data collected, and subjects involved. These records will need to be available upon request during the conduct of the study and retained by the university for three years after the conclusion of the study. If changes occur in recruiting of subjects, informed consent process or in your research protocol, or if unanticipated problems should arise it is the Researchers responsibility to notify the Office of Research or IRB in writing. The project should be discontinued until modifications can be reviewed and approved.

If you have any questions, please contact Dr. Mary Livingston at 257-4315.

A MEMBER OF THE UNIVERSITY OF LOUISIANA SYSTEM

P.O. BOX 3092 • RUSTON, LA 71272 • TELEPHONE (318) 257-5075 • FAX (318) 257-5079
AN EQUAL OPPORTUNITY UNIVERSITY



LOUISIANA TECH
UNIVERSITY

OFFICE OF UNIVERSITY RESEARCH

MEMORANDUM

TO: Dr. Vir Phoha

FROM: Barbara Talbot, University Research

SUBJECT: Human Use Committee Review

DATE: March 1, 2010

RE: Approved Continuation of Study HUC 416
Changing Number of Subjects from 500 to 2000

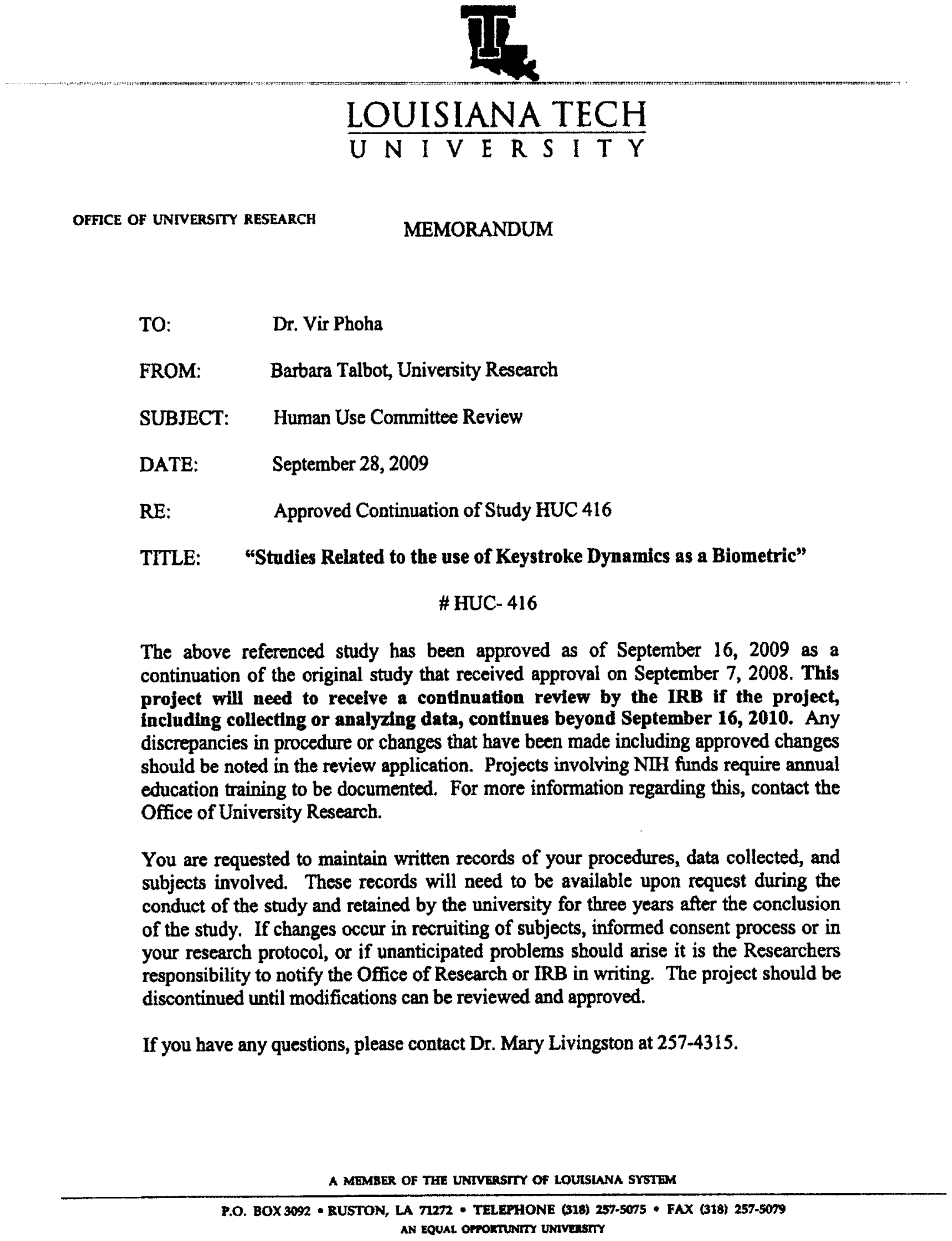
TITLE: **"Studies Related to the use of Keystroke Dynamics as a Biometric"**

HUC- 416

The above referenced study has been approved as of March 1, 2011 as a continuation of the original study that received approval on September 7, 2008. **This project will need to receive a continuation review by the IRB if the project, including collecting or analyzing data, continues beyond March 1, 2012.** Any discrepancies in procedure or changes that have been made including approved changes should be noted in the review application. Projects involving NIH funds require annual education training to be documented. For more information regarding this, contact the Office of University Research.

You are requested to maintain written records of your procedures, data collected, and subjects involved. These records will need to be available upon request during the conduct of the study and retained by the university for three years after the conclusion of the study. If changes occur in recruiting of subjects, informed consent process or in your research protocol, or if unanticipated problems should arise it is the Researchers responsibility to notify the Office of Research or IRB in writing. The project should be discontinued until modifications can be reviewed and approved.

If you have any questions, please contact Dr. Mary Livingston at 257-4315.



LOUISIANA TECH
UNIVERSITY

OFFICE OF UNIVERSITY RESEARCH

MEMORANDUM

TO: Dr. Vir Phoha

FROM: Barbara Talbot, University Research

SUBJECT: Human Use Committee Review

DATE: September 28, 2009

RE: Approved Continuation of Study HUC 416

TITLE: **"Studies Related to the use of Keystroke Dynamics as a Biometric"**

HUC- 416

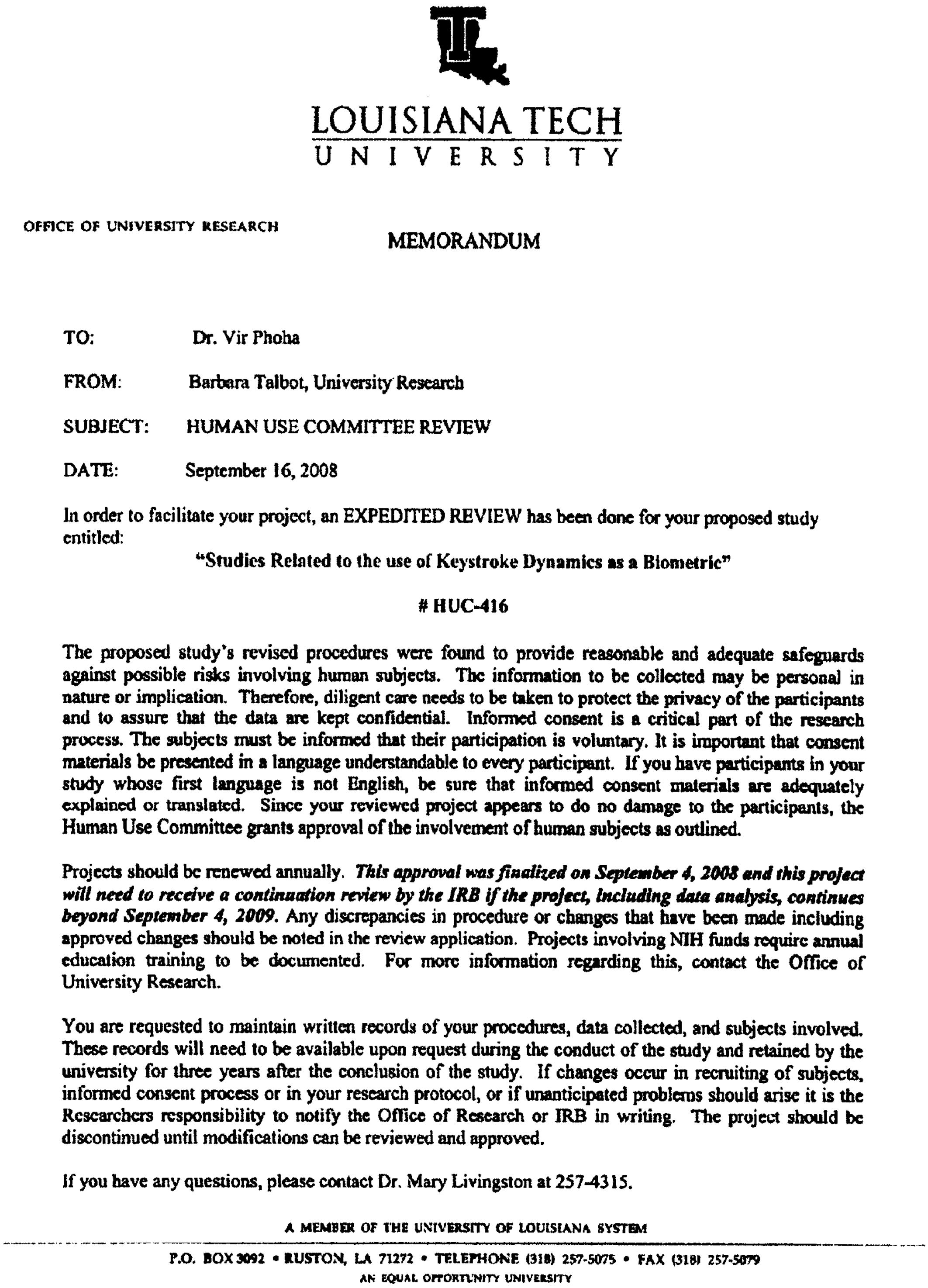
The above referenced study has been approved as of September 16, 2009 as a continuation of the original study that received approval on September 7, 2008. **This project will need to receive a continuation review by the IRB if the project, including collecting or analyzing data, continues beyond September 16, 2010.** Any discrepancies in procedure or changes that have been made including approved changes should be noted in the review application. Projects involving NIH funds require annual education training to be documented. For more information regarding this, contact the Office of University Research.

You are requested to maintain written records of your procedures, data collected, and subjects involved. These records will need to be available upon request during the conduct of the study and retained by the university for three years after the conclusion of the study. If changes occur in recruiting of subjects, informed consent process or in your research protocol, or if unanticipated problems should arise it is the Researchers responsibility to notify the Office of Research or IRB in writing. The project should be discontinued until modifications can be reviewed and approved.

If you have any questions, please contact Dr. Mary Livingston at 257-4315.

A MEMBER OF THE UNIVERSITY OF LOUISIANA SYSTEM

P.O. BOX 3092 • RUSTON, LA 71272 • TELEPHONE (318) 257-5075 • FAX (318) 257-5079
AN EQUAL OPPORTUNITY UNIVERSITY



LOUISIANA TECH
UNIVERSITY

OFFICE OF UNIVERSITY RESEARCH

MEMORANDUM

TO: Dr. Vir Phoha

FROM: Barbara Talbot, University Research

SUBJECT: HUMAN USE COMMITTEE REVIEW

DATE: September 16, 2008

In order to facilitate your project, an EXPEDITED REVIEW has been done for your proposed study entitled:

"Studies Related to the use of Keystroke Dynamics as a Biometric"

# HUC-416

The proposed study's revised procedures were found to provide reasonable and adequate safeguards against possible risks involving human subjects. The information to be collected may be personal in nature or implication. Therefore, diligent care needs to be taken to protect the privacy of the participants and to assure that the data are kept confidential. Informed consent is a critical part of the research process. The subjects must be informed that their participation is voluntary. It is important that consent materials be presented in a language understandable to every participant. If you have participants in your study whose first language is not English, be sure that informed consent materials are adequately explained or translated. Since your reviewed project appears to do no damage to the participants, the Human Use Committee grants approval of the involvement of human subjects as outlined.

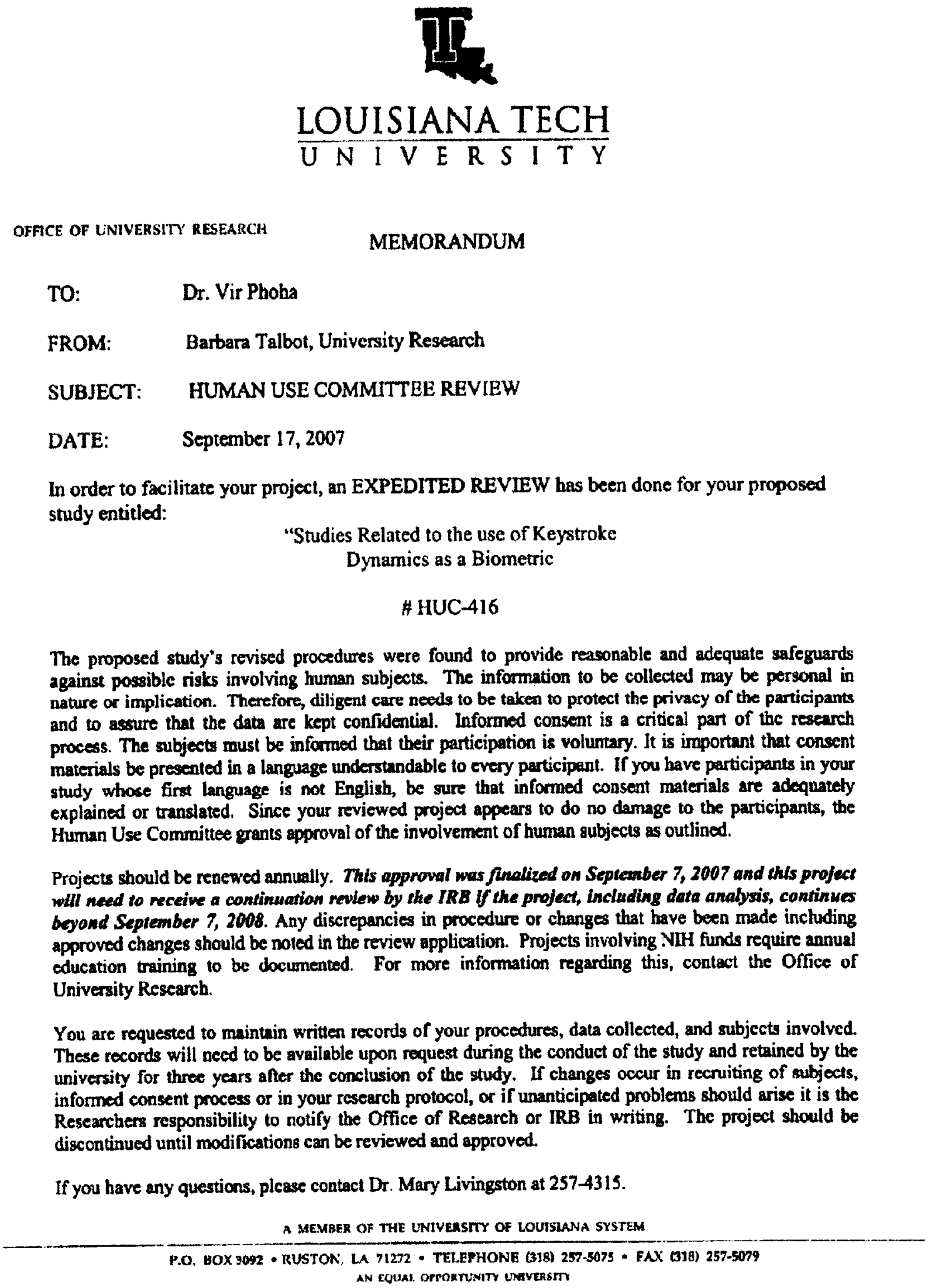
Projects should be renewed annually. ***This approval was finalized on September 4, 2008 and this project will need to receive a continuation review by the IRB if the project, including data analysis, continues beyond September 4, 2009.*** Any discrepancies in procedure or changes that have been made including approved changes should be noted in the review application. Projects involving NIH funds require annual education training to be documented. For more information regarding this, contact the Office of University Research.

You are requested to maintain written records of your procedures, data collected, and subjects involved. These records will need to be available upon request during the conduct of the study and retained by the university for three years after the conclusion of the study. If changes occur in recruiting of subjects, informed consent process or in your research protocol, or if unanticipated problems should arise it is the Researchers responsibility to notify the Office of Research or IRB in writing. The project should be discontinued until modifications can be reviewed and approved.

If you have any questions, please contact Dr. Mary Livingston at 257-4315.

A MEMBER OF THE UNIVERSITY OF LOUISIANA SYSTEM

P.O. BOX 3092 • RUSTON, LA 71272 • TELEPHONE (318) 257-5075 • FAX (318) 257-5079
AN EQUAL OPPORTUNITY UNIVERSITY



LOUISIANA TECH
UNIVERSITY

OFFICE OF UNIVERSITY RESEARCH

MEMORANDUM

TO: Dr. Vir Phoha

FROM: Barbara Talbot, University Research

SUBJECT: HUMAN USE COMMITTEE REVIEW

DATE: September 17, 2007

In order to facilitate your project, an EXPEDITED REVIEW has been done for your proposed study entitled:

"Studies Related to the use of Keystroke
Dynamics as a Biometric

HUC-416

The proposed study's revised procedures were found to provide reasonable and adequate safeguards against possible risks involving human subjects. The information to be collected may be personal in nature or implication. Therefore, diligent care needs to be taken to protect the privacy of the participants and to assure that the data are kept confidential. Informed consent is a critical part of the research process. The subjects must be informed that their participation is voluntary. It is important that consent materials be presented in a language understandable to every participant. If you have participants in your study whose first language is not English, be sure that informed consent materials are adequately explained or translated. Since your reviewed project appears to do no damage to the participants, the Human Use Committee grants approval of the involvement of human subjects as outlined.

Projects should be renewed annually. ***This approval was finalized on September 7, 2007 and this project will need to receive a continuation review by the IRB if the project, including data analysis, continues beyond September 7, 2008.*** Any discrepancies in procedure or changes that have been made including approved changes should be noted in the review application. Projects involving NIH funds require annual education training to be documented. For more information regarding this, contact the Office of University Research.

You are requested to maintain written records of your procedures, data collected, and subjects involved. These records will need to be available upon request during the conduct of the study and retained by the university for three years after the conclusion of the study. If changes occur in recruiting of subjects, informed consent process or in your research protocol, or if unanticipated problems should arise it is the Researchers responsibility to notify the Office of Research or IRB in writing. The project should be discontinued until modifications can be reviewed and approved.

If you have any questions, please contact Dr. Mary Livingston at 257-4315.

BIBLIOGRAPHY

[1] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognition*, vol. 38, no. 12, pp. 2270 – 2285, December 2005.

[2] A. Ross, K. Nandakumar, and A. Jain, *Handbook of Biometrics.* Springer-Verlag, 2006.

[3] A. Ross and A. Jain, "Information fusion in biometrics," *Pattern Recognition Letters*, vol. 24, pp. 2115–2125, 2003.

[4] S. Prabhakar and A. K. Jain, "Decision-level fusion in fingerprint verification," *Pattern Recognition*, vol. 35, no. 4, pp. 861 – 874, 2002.

[5] K.-A. Toh, X. Jiang, and W.-Y. Yau, "Exploiting global and local decisions for multimodal biometrics verification," *Signal Processing, IEEE Transactions on*, vol. 52, no. 10, pp. 3059–3072, October 2004.

[6] R. Brunelli and D. Falavigna, "Person identification using multiple cues," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 10, pp. 955–966, October 1995.

[7] B. Ulery, A. Hicklin, C. Watson, W. Fellner, and P. Hallinan, "Studies of biometric fusion–executive summary," National Institute of Standards and Technology, Tech. Rep. 7346, September 2006.

[8] A. Kumar and D. Zhang, "Personal recognition using hand shape and texture," *Image Processing, IEEE Transactions on*, vol. 15, no. 8, pp. 2454–2461, August 2006.

[9] K. Nandakumar, Y. Chen, S. C. Dass, and A. Jain, "Likelihood ratio-based biometric score fusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 342–347, 2008.

[10] M. He, S. Horng, P. Fan, R. Run, R. Chen, J. Lai, M. Khan, and K. Sentosa, "Performance evaluation of score level fusion in multimodal biometric systems," *Pattern Recognition*, vol. 43, no. 5, pp. 1789 – 1800, 2010.

[11] F. Yang and B. Ma, "A new mixed-mode biometrics information fusion based-on fingerprint, hand-geometry and palm-print," in *Image and Graphics, 2007. ICIG 2007. Fourth International Conference on*, pp. 689–693, August 2007.

[12] M. Monwar and M. Gavrilova, "Fes: A system for combining face, ear and signature biometrics using rank level fusion," in *Information Technology: New Generations, 2008. Fifth International Conference on*, pp. 922–927, April 2008.

[13] L. Nanni and A. Lumini, "Ensemble of multiple palmprint representation," *Expert Systems with Applications*, vol. 36, no. 3, Part 1, pp. 4485 – 4490, 2009.

[14] E. Marasco and C. Sansone, "Improving the accuracy of a score fusion approach based on likelihood ratio in multimodal biometric systems," in *Image Analysis and Processing*, ser. LNCS, vol. 5716, pp. 509–518, 2009.

[15] A. Lumini and L. Nanni, "When fingerprints are combined with iris - a case study: Fvc2004 and casia." *International Journal of Network Security*, vol. 4, no. 1, pp. 27–34, 2007.

[16] J. Fierrez-Aguilar, L. Nanni, J. Ortega-Garcia, R. Cappelli, and D. Maltoni, "Combining multiple matchers for fingerprint verification: a case study in fvc2004," in *Proc. 13th IAPR Intl. Conf. on Image Analysis and Processing, ICIAP*, ser. LNCS, vol. 3617. Springer, pp. 1035–1042, September 2005.

[17] C. Bergamini, L. Oliveira, A. Koerich, and R. Sabourin, "Combining different biometric traits with one-class classification," *Signal Processing*, vol. 89, no. 11, pp. 2117 – 2127, 2009.

[18] R. Snelick, U. Uludag, A. Mink, M. Indovina, and A. Jain, "Large scale evaluation of multimodal biometric authentication using state-of-the-art systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 450–455, 2005.

[19] J. Fierrez-Aguilar, J. Ortega-Garcia, J. Gonzalez-Rodriguez, and J. Bigun, "Discriminative multimodal biometric authentication based on quality measures," *Pattern Recognition*, vol. 38, no. 5, pp. 777 – 779, 2005.

[20] Y. Ma, B. Cukic, and H. Singh, "A classification approach to multi-biometric score fusion," in *Audio- and Video-Based Biometric Person Authentication*, ser. Lecture Notes in Computer Science. Springer, vol. 3546, pp. 484–493, 2005.

[21] P. Griffin, "Optimal biometric fusion for identity verification," Identix Research, Tech. Rep. RDNJ-03-0064, 2004.

[22] C. Sansone and M. Vento, "Signature verification: Increasing performance by a multi-stage system," *Pattern Analysis and Applications*, vol. 3, pp. 169–181, 2000.

[23] G. L. Marcialis, F. Roli, and L. Didaci, "Personal identity verification by serial fusion of fingerprint and face matchers," *Pattern Recognition*, vol. 42, no. 11, pp. 2807 - 2817, 2009.

[24] G. Marcialis, P. Mastinu, and F. Roli, "Serial fusion of multi-modal biometric systems," in *Biometric Measurements and Systems for Security and Medical Applications (BIOMS), 2010 IEEE Workshop on*, pp. 1 –7, September 2010.

[25] Z. Akhtar, G. Fumera, G. Marcialis, and F. Roli, "Evaluation of serial and parallel multibiometric systems under spoofing attacks," in *Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*, pp. 283 –288, September 2012.

[26] L. Allano, B. Dorizzi, and S. Garcia-Salicetti, "Tuning cost and performance in multi-biometric systems: A novel and consistent view of fusion strategies based on the sequential probability ratio test (sprt)," *Pattern Recogn. Lett.*, vol. 31, no. 9, pp. 884–890, July 2010.

[27] K. Takahashi, M. Mimura, Y. Isobe, and Y. Seto, "A secure and user-friendly multimodal biometric system," *Proceedings of the SPIE*, vol. 5404, pp. 12–19, 2004.

[28] T. Murakami, K. Takahashi, and K. Matsuura, "Towards optimal countermeasures against wolves and lambs in biometrics," in *Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*, pp. 69 –76, September 2012.

[29] C. K. Chow, "On optimum recognition error and reject tradeoff," *Information Theory, IEEE Transactions on*, vol. 16, no. 1, pp. 41–46, January 1970.

[30] G. Fumera, F. Roli, and G. Giacinto, "Reject option with multiple thresholds," *Pattern Recognition*, vol. 33, pp. 2099–2101, 2000.

[31] F. Tortorella, "An optimal reject rule for binary classifiers," in *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition.* London, UK: Springer-Verlag, pp. 611–620, 2000.

[32] C. M. Santos-Pereira and A. M. Pires, "On optimal reject rules and roc curves," *Pattern Recogn. Lett.*, vol. 26, pp. 943–952, May 2005.

[33] P. L. Bartlett and M. H. Wegkamp, "Classification with a reject option using a hinge loss," *J. Mach. Learn. Res.*, vol. 9, pp. 1823–1840, June 2008.

[34] M. Hossain, K. Balagani, and V. Phoha, "New impostor score based rejection methods for continuous keystroke verification with weak templates," in *Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*, pp. 251–258, 2012.

[35] T. C. W. Landgrebe, D. M. J. Tax, P. Paclík, and R. P. W. Duin, "The interaction between classification and reject performance for distance-based reject-option classifiers," *Pattern Recogn. Lett.*, vol. 27, pp. 908–917, June 2006.

[36] L. Allano, S. Garcia-Salicetti, and B. Dorizzi, "An adaptive multi-biometric incremental fusion strategy in the context of bmec 2007," in *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pp. 1144–1149, December 2008.

[37] D. Gunetti and C. Picardi. Keystroke analysis of free text. *ACM Trans. Inf. Syst. Secur.*, 8(3):312–347, August 2005.

[38] K. Killourhy and R. Maxion. Why did my detector do that?! predicting keystroke-dynamics error rates. In *Recent Adv. in Intrusion Detection*, pages 256–276, Canada, 2010.

[39] E. Yu and S. Cho. Ga-svm wrapper approach for feature subset selection in keystroke dynamics identity verification. In *Intl. Joint Conf. on Neu. Nets.*, pages 2253–2257, 2003.

[40] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. on SMC*, 9(1):62–66, 1979.

[41] "Biometric scores set," November 7, 2011, retrieved January 17, 2014, from Information Technology Laboratory, The National Institute of Standards and Technology (NIST): http://www.nist.gov/itl/iad/ig/biometricscores.cfm.

[42] Q. Tao and R. Veldhuis, "Hybrid fusion for biometrics: Combining score-level and decision-level fusion," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Workshop on Biometrics*, pp. 1–6, June 2008.

[43] S. Dass, K. Nandakumar, and A. Jain, "A principled approach to score level fusion in multimodal biometric systems," in *Audio- and Video-Based Biometric Person Authentication*, ser. Lecture Notes in Computer Science. Springer, vol. 3546, pp. 1049–1058, 2005.

[44] B. Dubuisson and M. Masson. A statistical decision rule with incomplete knowledge about classes. *Pattern Recognition*, 26(1):155 - 165, 1993.

[45] M. Hossain, K. Balagani, and V. Phoha, "On controlling genuine reject rate in multi-stage biometric verification," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pp. 194–199, June 2013.

[46] K. Rahman, K. Balagani, and V. Phoha, "Snoop-forge-replay attacks on continuous verification with keystrokes," *Information Forensics and Security, IEEE Transactions on*, vol. 8, no. 3, pp. 528–541, March 2013.

[47] K. Rahman, K. Balagani, and V. Phoha, "Making impostor pass rates meaningless: A case of snoop-forge-replay attack on continuous cyber-behavioral verification with keystrokes," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Conference on*, June 2011, pp. 31–38.

[48] V. Phoha and S. Joshi, "Method and system of Identifying users based upon free text keystroke," US Patent Number 8489635, 2013.

[49] A. Wald, *Sequential Analysis*, 1st ed. John Wiley and Sons, 1947.

[50] M. Girolami and C. He, "Probability density estimation from optimally condensed data samples," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 10, pp. 1253–1264, October 2003.

[51] S. Prabhakar and A. K. Jain, "Decision-level fusion in fingerprint verification," *Pattern Recognition*, vol. 35, pp. 861–874, 2001.

[52] S. Dass, Y. Zhu, and A. Jain, "Validating a biometric authentication system: Sample size requirements," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 12, pp. 1902 –1319, December 2006.

[53] K. Killourhy and R. Maxion. Comparing anomaly-detection algorithms for keystroke dynamics. In *Intl. Conf. on Dependable Systems and Networks*, pages 125–134, 2009.

[54] F. Bergadano, D. Gunetti, and C. Picardi. User authentication through keystroke dynamics. *ACM Trans. Inf. Syst. Secur.*, 5:367–397, November 2002.

[55] J. Ilonen, "Keystroke dynamics," *Advanced Topics in Information Processing - Lecture*, 2003.

[56] A. Kumar, "Incorporating cohort information for reliable palmprint authentication," in *Computer Vision, Graphics Image Processing, 2008. ICVGIP '08. Sixth Indian Conference on*, pp. 583 –590, December 2008.

[57] M. E. Schuckers, "Using the beta-binomial distribution to assess performance of a biometric identification device," *International Journal of Image and Graphics*, vol. 3, no. 3, pp. 523–529, July 2003.

[58] A. Wald, "Sequential tests of statistical hypotheses," *The Annals of Mathematical Statistics*, vol. 16, no. 2, pp. 117–186, 1945.

[59] J. Q. Li and A. R. Barron, "Mixture density estimation," *in Advances in Neural Information Processing Systems 12*, pp. 279–285, 1999.

[60] A. Rakhlin, D. Panchenko, and S. Mukherjee, "Risk bounds for mixture density estimation," *ESAIM: Probability and Statistics*, vol. 9, pp. 220–229, October 2005.

[61] J. Huang and C. Ling, "Using auc and accuracy in evaluating learning algorithms," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 3, pp. 299–310, 2005.

[62] H. A. Guvenir and M. Kurtcephe, "Ranking instances by maximizing the area under roc curve," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 10, pp. 2356–2366, 2013.

[63] R. Wang and K. Tang, "Feature selection for maximizing the area under the roc curve," in *Data Mining Workshops, 2009. ICDMW '09. IEEE International Conference on*, pp. 400–405, 2009.

[64] C. Léger and J. Romano, "Bootstrap choice of tuning parameters," *Annals of the Institute of Statistical Mathematics*, vol. 42, pp. 709–735, 1990.

[65] T. Scheidat, A. Engel, and C. Vielhauer, "Parameter optimization for biometric fingerprint recognition using genetic algorithms," in *Workshop on Mult. Sec.*, NY, pp. 130–134, 2006.

[66] "Intel 64 and ia-32 architectures software developer's manual, volume 2, instruction set reference, a-z," February, 2014, retrieved April 15, 2014, from Intel: http://www.intel.com/content/dam/www/public/us/en/ documents/ manuals/ 64-ia-32-architectures-software-developer-instruction-set-reference-manual-325383.pdf.

[67] "Processor instruction timings," retrieved April 15, 2014, from ARM: http://infocenter.arm.com/help/index.jsp?topic=/ com.arm.doc.ddi0337e/ BABBCJII.html.

[68] J. Fierrez-Aguilar, J. Ortega-Garcia, and J. Gonzalez-Rodriguez. Target dependent score normalization techniques and their application to signature verification. *IEEE Trans. on SMC-Part C*, 35(3):418 –425, 2005.

[69] N. Poh, T. Bourlai, J. Kittler, L. Allano, F. Alonso-Fernandez, O. Ambekar, J. Baker, B. Dorizzi, O. Fatukasi, J. Fierrez, H. Ganster, J. Ortega-Garcia, D. Maurer, A. A. Salah, T. Scheidat, and C. Vielhauer. Benchmarking quality-dependent and cost-sensitive score-level multimodal biometric fusion algorithms. *Trans. Info. For. Sec.*, 4(4):849–866, December 2009.

[70] H. A. David. *Ordered Statistics.* Wiley, 1981.

[71] A. Sarma and D. Tufts. Robust adaptive threshold for control of false alarms. *IEEE Sig. Proc. Let.*, 8(9):261 –263, 2001.

[72] K. Balagani, V. Phoha, A. Ray, and S. Phoha. On the discriminability of keystroke feature vectors used in fixed text keystroke authentication. *Pattern Recog. Let.*, 32(7), 2011.