January 2014

Hybrid Data Storage Framework for the Biometrics Domain

Abhinav Tiwari
The University of Western Ontario

Supervisor
Dr. Miriam A.M. Capretz
The University of Western Ontario

Graduate Program in Electrical and Computer Engineering

A thesis submitted in partial fulfillment of the requirements for the degree in Master of Engineering Science

HYBRID DATA STORAGE FRAMEWORK FOR THE BIOMETRICS DOMAIN

By

Abhinav Tiwari

Graduate Program in Engineering Science
Department of Electrical and Computer Engineering

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Engineering Science

The School of Graduate and Postdoctoral Studies
Western University
London, Ontario, Canada

Abstract

Biometric based authentication is one of the most popular techniques adopted in large-scale identity matching systems due to its robustness in access control. In recent years, the number of enrolments has increased significantly posing serious issues towards the performance and scalability of these systems. In addition, the use of multiple modalities (such as face, iris and fingerprint) is further increasing the issues related to scalability. This research work focuses on the development of a new Hybrid Data Storage Framework (HDSF) that would improve scalability and performance of biometric authentication systems (BAS). In this framework, the scalability issue is addressed by integrating relational database and NoSQL data store, which combines the strengths of both. The proposed framework improves the performance of BAS in three areas (i) by proposing a new *biographic match score based key filtering* process, to identify any duplicate records in the storage (de-duplication search); (ii) by proposing a *multi-modal biometric index based key filtering* process for identification and de-duplication search operations; (iii) by adopting parallel biometric matching approach for identification, enrolment and verification processes. The efficacy of the proposed framework is compared with that of the traditional BAS and on several values of False Rejection Rate (FRR). Using our dataset and algorithms it is observed that when compared to traditional BAS, the HDSF is able to show an overall efficiency improvement of more than 54% for zero FRR and above 60% for FRR values between 1-3.5% during identification search operations.

Keywords

Biometric Authentication System, Biometric Technology, Large-Scale Identity Matching, Biographic Fusion, Scalable Biometric Systems, NoSQL Biometric Storage, Parallel Biometric Matching, Identification Search, De-duplication Search

Acknowledgments

This thesis may have only one name listed as the author, but it could not have been written without the assistance and guidance of several people. First and foremost, I offer my sincere gratitude to my supervisor, Prof. Miriam A. M. Capretz, Associate Professor in the Department of Electrical and Computer Engineering at the Western University. Dr. Capretz has been a source of guidance, encouragement and support throughout this process. Dr. Capretz has greatly impacted this work, and her effort, time and dedication will always be appreciated. Dr. Capretz supported me throughout my thesis with her keen observations, patience and immense knowledge whilst allowing me the room to work in my own way.

To my father, I would like to express my deepest gratitude and love. I would not be in the position to write this thesis without his emotional, spiritual and technical support. For always having faith in me and allowing me to be as ambitious as I wanted, knowing he will be there in my success and failures, I would like to dedicate this work to him.

Dr. Shuying Vinson Wang, Katarina Grolinger and Wilson Higashino are all the team members and colleagues who deserve special recognition for their valued inputs which led directly to the success of this work. To the many professors, teachers and others who have helped educate me both in regard to this thesis and in general, I would also like to thank them all.

Lastly I would like to thank my family and friends, all of whom have contributed greatly to shaping me throughout my life.

Table of Contents

List of Tables

List of Figures

List of Appendices

List of Abbreviations

API	Application Programming Interface
BAS	Biometric Authentication System
BBM	Biometric Biographic Management
BDDI	Biographic Data of Duplicate Identity
BFS	Biographic Fused Score
BIS	Biometric Image Set
BMS	Biographic Match Score
BTS	Biometric Template Set
DMA	Direct Memory Access
EER	Equal Error Rate
FAR	False Acceptance Rate
FMR	False Match Rate
FNMR	False Non-Match Rate
FRR	False Rejection Rate
GAR	Genuine Acceptance Rate
GRR	Genuine Rejection Rate
HDSF	Hybrid Data Storage Framework
HPC	High Performance Computing
JSON	JavaScript Object Notation
TEA	Template Extraction Algorithm
TMA	Template Matching Algorithm
ME	Match Engine
NDDS	NoSQL Distributed Data Storage
NoSQL	Not only SQL
RAM	Random Access Memory
RDBMS	Relational Database Management System
SQL	Structured Query Language
WCF	Windows Communication Foundation
XML	eXtensible Markup Language

Chapter 1

1 Introduction

Biometric based authentication serves as the underlying technology for modern Access Control Systems [1]. An access control system ensures that a user possesses selective privilege towards what a user can access physically or through a program executing on behalf of the user [2], in a resource. Therefore, an access control system requires a robust authentication mechanism to verify the identity of the users and provide access to the authorized users while rejecting access to an impostor. Biometric Authentication fulfills this requirement by offering multiple levels of performance and security which could protect resources such as buildings, railway stations or airports; and logical access control systems such as computers, cellphones and ATMs [1], [3], [4].

A Biometric Authentication System (BAS) captures the biometric images pertaining to different modalities (e.g. face, iris, fingerprints) and sub-modalities (e.g. left-iris, right-iris, left-index-finger, right-index-finger) of a user, and converts these images to biometric templates. These biometric templates are matched with templates stored in the system to come up with a match/no-match decision. Traditional approaches using biometrics had manual or semi-automated approach for authentication, working only on textual data along with manual inspection of face or fingerprint images [5]. On the other hand, a BAS automates the process of authentication by using biometric devices and algorithms, providing much higher accuracy over manual or semi-automated approaches [6]. BAS advantages come from the fact that BAS capture the biometric data of a user

through sensors and provide a match/no-match decision after comparing it with thousands of records in a short time. Therefore, the accuracy and performance offered by BAS is much higher than those achieved in manual or semi-automated approaches.

The benefits of using BAS for human recognition and authentication process has, resulted in their increased adoption in large number of authentication systems across the globe. Moreover, its importance in modern times is strengthened by the need for large-scale identity matching systems in several application domains such as healthcare, banking, insurance, government welfare schemes and border control [3].

1.1 Motivation

The benefits of using BAS for human recognition and authentication come along with several challenges [7], where scalability and performance are major areas of concern. As establishing the identity of a user with high confidence is becoming critical in our vastly interconnected society [3], BAS are being increasingly adopted in large number of applications. Not only small applications used inside an organization or a group of organizations, larger systems used by government and national agencies for applications such as national ID card, social security, e-passport systems, border control, welfare disbursement, have also started leveraging the benefits of biometric technology. These larger systems deal with the biometric and biographic data of several millions of users, where the number of users is increasing each day. For example, the biometric database of the US-Visit [8] contains millions of records and has grown from 4.5 TB in 2007 to 7 TB in the year of 2010, where its size is still increasing. Similarly, the Aadhaar scheme [9] offered in India aiming to offer unique identification number to their citizens, is supposed

to store data for 1.2 billion identities in its initial plan, which will further grow in size due to the increasing population. These large number of enrolments result in generating massively huge datasets comprising of both biographic and biometric data in the scale of gigabytes and terabytes respectively. Due to the continued growth of these datasets, existing biometrics systems are reaching their limits of scalability. Existing biometric systems based on traditional storage approaches are currently incapable of handling these massive datasets due to an adverse impact of scalability on accuracy of the system [7].

Furthermore, multimodal biometric systems have seen an increase in their adoption due to their higher performance over unimodal systems [10], [11]. Most of the existing biometric systems use more than one biometric modality in order to achieve higher accuracy and higher throughput. For example, the US-Visit [8] database containing fingerprints and face images; Aadhaar scheme [9] using fingerprint, face and iris; and the FBI's Next Generation Identification System [12] incorporating fingerprint, face, iris and palmprint; are all multi-modal biometric authentication systems. With an increase in the number of modalities, the amount of data pertaining to biometric images and templates related to different modalities is increasing rapidly. Furthermore, most of these systems store more than one biometric image and template for each biometric modality and sub-modality for improving recognition accuracy and the overall performance of the system [13]. Also, some systems perform multiple enrolments (which is a process of storing biometric and biographic data of a user in the system) for the same reason. For example, the Aadhaar scheme performs enrolments in two sessions and stores multiple face, iris and fingerprint images in each session, to improve recognition accuracy [14]. Therefore, using a multi-modal system further raises the scalability issues in BAS to a large extent.

Due to the above mentioned factors, the biometric datasets today have grown too big to be managed and processed by traditional data storage technologies [8], [9]. Today, biometric datasets are facing the same issues often associated with the term “Big data”, due to their large size and the requirements of achieving faster recognition rate in biometric systems for current applications [15]. Therefore, it is an important challenge to define an effective data storage strategy which could be used by large-scale biometric datasets and provides horizontal scalability.

Similarly to scalability, performance is another major bottleneck in large-scale biometric systems. A biometric identification and de-duplication (which is done to check duplicate biometric records in the system) search operation in BAS requires matching an input biometric data with all the data stored in the system. As with the increasing number of enrolments in most of the existing biometric systems, the size of the stored biometric data has become huge. Therefore, the biometric identification and de-duplication search operations consume a significant amount of time resulting in an approach to perform them as offline operations [9], [16]. This introduces an additional problem of multiple enrolments of a user during an enrolment process. A delay in performing de-duplication could result in providing unauthorized access to a resource resulting in loss of security. Moreover, a BAS often connected with multiple client applications sending biometric verification requests, requires adopting methodologies which could help in serving those requests simultaneously. Therefore, improvement in performance of a biometric system also needs careful attention together with its scalability.

1.2 Contributions

There are several contributions that together form the scope of this thesis, which will now be outlined. The primary offering is the creation of a Hybrid Data Storage Framework (HDSF), which in turn provides the following contributions:

- It provides a horizontally scalable storage for large-scale identity matching applications in order to store large biometric datasets, as opposed to a traditional RDBMS [17] based storage providing vertical scalability or a memory based storage limited towards its size.
- It provides mechanisms for making the de-duplication and identification processes, an online operation as opposed to the traditional systems performing them offline. Authenticating a user in online mode will eliminate the risks of multiple enrolments and consequently remove the threat to the security of the system.
- It provides mechanisms for processing multiple verification requests simultaneously.

In addition to providing contributions related to achieving horizontal scalability and higher performance, HDSF also provides the following enhancements over traditional BAS:

- It provides on-the-fly selection of different biometric algorithms based on different application requirements in terms accuracy versus efficiency trade-off, which is important for the adoption of framework by a vast number of applications.

- It provides a biometric modality independent framework, as opposed to most of the existing systems which are bound to a specific modality [18], [19]. HDSF supports multiple modalities and does not present any limitation towards their number and type.
- It provides capability to store biographic data and provide querying based on it, which is a major limitation in a number of existing file system based [18]–[20] and memory based [21] approaches.
- It provides functionalities related to biometric systems through an Application Programming Interface (API). Moreover, this API is exposed as a service in order to enable access through different applications and devices.

1.3 Organization of the Thesis

This thesis is organized into chapters as follows:

- Chapter 2 contains an introduction to Biometric Authentication Systems (BAS). The key concepts behind BAS, its sub-systems and biometric algorithms are discussed in detail. Further, the different operation modes of BAS and associated performance metrics are explained, followed by a discussion of related work in biometrics domain.
- Chapter 3 focuses on existing data storage technologies in order to identify the suitable storage for each of the different types of data pertaining to biometrics domain. The key features of Relational DBMS are discussed and examined in order to analyze its suitability as a data storage option. Further, different categories of NoSQL data stores are thoroughly investigated in order to evaluate their fitness in biometrics domain.

- Chapter 4 proposes a Hybrid Data Storage Framework (HDSF) in order to provide scalable storage and providing performance improvement during the course of performing different biometric processes. The different layers of HDSF and their internal sub-systems are discussed in detail in order to highlight the key improvements in HDSF over traditional Biometric Authentication Systems.
- Chapter 5 provides details about the implementation and evaluation of the proposed Hybrid Data Storage framework. The different biometric algorithms, datasets, and storage technologies used during the evaluation of HDSF are presented. Finally, a detailed explanation of the different levels of performance improvement achieved by HDSF and their effect on different biometric processes is presented.
- Chapter 6 presents the conclusion to the thesis, as well as outlines the possibilities for future work.

Chapter 2

2 Background and Related Work

This chapter lays the foundation for the biometrics domain and discusses the concepts behind biometric authentication technology in section 2.1. As a next step, the different types of datasets and biometric algorithms related to biometrics domain are discussed in sections 2.2 and 2.3, respectively. In section 2.4, the model of a typical Biometric Authentication System (BAS) has been presented and its different sub-systems are mentioned in detail. Further, the different operating modes associated with a BAS and its performance metrics are discussed in section 2.5 and 2.6, respectively. Finally, the related work describing the existing approaches addressing the needs of biometrics domain are discussed thoroughly in section 2.7, highlighting their key aspects and their bottlenecks specifically in terms of handling large biometric datasets.

2.1 Biometric Authentication Technology

Biometrics is the science of establishing the identity of a user based on the physical (face, iris, fingerprint), chemical (DNA) or behavioral (gait, signature, keystroke) attributes of the user [22][23]. These physical, chemical or behavioral characteristics of different users are termed as biometric modalities and are unique to different users. Moreover, any human physiological and/or behavioral characteristic can be used as a biometric characteristic, as long as it satisfies the requirements of universality, distinctiveness, permanence and collectability [24]. These characteristics of biometrics are collectively exploited in a BAS. Some of the most common biometric modalities [25] are mentioned

in Table 2.1. However, the number of modalities is increasing and some new modalities such as lip-print have also been introduced recently [26]. Furthermore, most of the biometric modalities are associated with two or more sub-modalities. For example, iris has two sub-modalities (left-iris and right-iris), whereas fingerprints have 10 sub-modalities (5 fingers for each left and right hands).

Table 2.1: Biometric Modalities

Physiological		**Behavioral**	**Other**
Face Image	Ear Shape	Keystroke	Odor
Fingerprint	Palmprint	Signature	DNA
Finger Vein	Hand Vein	Gait	ECG
Finger Geometry	Hand Geometry	Speech	Hand Thermogram
Retinal Scan	Iris Scan	Voice	Facial Thermogram

Biometric based authentication provides certain advantages over traditional schemes based on passwords and tokens [5], such as:

1. **Negative Recognition:** Negative recognition ensures that a user is not enrolled in a system multiple times under different identities. In case when the same user is enrolled multiple times, he/she can exploit the system by using it more than once even when not authorized to do so. For example, a user could attempt to claim multiple benefits under different names from a government offered welfare scheme.
2. **Non-Repudiation:** Non-Repudiation ensures that a user who accessed a particular system cannot later deny by claiming that an impostor might have used the system.

This is common in the case of passwords and tokens that a user can later claim that his/her credentials were stolen and used by an impostor.

2.2 Biometric and Biographic Datasets

A biometric dataset belonging to different users consists of either or both of a Biometric Image Set (BIS) and a Biometric Template Set (BTS) [27]. Moreover, a biometric dataset is often associated with a Biographic Dataset (BD), used to store other information about a user [28][22]. The three types of datasets mentioned above could be defined as the following:

Definition 2.2.1 (Biometric Image Set): A Biometric Image Set (BIS) consists of the biometric images for a user [29]. Therefore, a BIS could be defined as a set of all biometric images such that $BIS = \{BIS_1^s, BIS_2^s, ..., BIS_i^s\}$ where each image BIS_i^s has a different sub-modality s. Here, s represents the biometric sub-modality such that $s \in S$ (Set of all biometric sub-modalities) and i represents the image count.

Definition 2.2.2 (Biometric Template Set): A Biometric Template Set (BTS) could be defined as a set of biometric templates such that $BTS = \{BTS_1^s, BTS_2^s, ..., BTS_i^s\}$ [23][29] where each template BTS_i^s has a different sub-modality s. Here, s represents the biometric sub-modality such that $s \in S$ (Set of all biometric sub-modalities) and i represents the template count. A biometric template consists of biometric features or patterns extracted out of the biometric image set BIS represented as binary information [24]. Similar to BIS, a BTS may contain more than one template belonging to a user with

each template representing a different sub-modality $s \in S$ (Set of all biometric sub-modalities).

Definition 2.2.3 (Biographic Dataset): A Biographic Dataset (BD) consists of different biographic information associated with a user such as name, personal identification number and address [22]. In order to correctly identify a user, this information should be stored inside a biometric authentication system along with the user's biometric information. However, the type of information included in a BD could vary from one biometric system to another.

2.3 Biometric Algorithms

Biometric algorithms provide automated methods that enable a biometric system to recognize a user by his or her biometric traits [24]. These methods consist of a series of steps often grouped into two major processes: template extraction and template matching. A biometric algorithm could combine them as a single process or may provide them as separate processes in two different algorithms: Template Extraction Algorithm (TEA) and Template Matching Algorithm (TMA) [27]. More often, the two processes are kept separate as template extraction is done only once for each new biometric input; whereas, a matching process is performed repeatedly whenever a new input template is matched with one or more stored templates. However, the implementation of biometric algorithms is beyond the scope of this thesis, the functionalities provided by the two categories of biometric algorithms are explained in the following sections in order to provide a better understanding of their roles inside a BAS explained in section 2.4.

2.3.1 Template Extraction Algorithms

The task of a template extraction algorithm is to extract biometric patterns or features from a raw biometric image. The resulting biometric pattern could be written to a binary file termed as a 'template', which could be either used for matching or is stored in the database for future matching purposes. Different biometric modalities such as face, fingerprint, and iris consist of different types of biometric features, requiring different algorithms to be used for extraction [30]. Therefore, a set of template extraction algorithms could be defined as $TEA = \{TEA_1^m, TEA_2^m, ..., TEA_i^m\}$ where each algorithm TEA_i^m is applied to a modality m such that $m \in M$ (Set of all biometric modalities) and i represents the algorithm count in TEA.

2.3.2 Template Matching Algorithms

The task of a template matching algorithm is to match biometric patterns or features written inside two biometric template files. A biometric matching could only be performed between two templates and not raw images; therefore, a template extraction is always carried out in case the input is a biometric image, in order to convert it to a template. Similar to extraction algorithms, different algorithms are required to match each of the different biometric modalities such as face, fingerprint, and iris [30]. For instance, an algorithm to match two face images will be different from that used to match two fingerprint templates. Therefore, a set of template matching algorithms could be defined as $TMA = \{TMA_1^m, TMA_2^m, ..., TMA_i^m\}$ where each algorithm TMA_i^m is applied to a modality m such that $m \in M$ (Set of all biometric modalities) and i represents the algorithm count in TMA.

2.4 Biometric Authentication Systems (BAS)

A Biometric Authentication System employs a biometric based authentication scheme to protect resources. As shown in Figure 2.1, a BAS typically consists of the following modules: input devices, template extractor, template matcher, match decision, storage and a single BAS controller [28]. The template matcher and match decision modules could be either implemented as separate modules providing matching and decision functionalities [31][32][10], or as a combined module [24][22]. The functionality of each BAS module is given as follows:

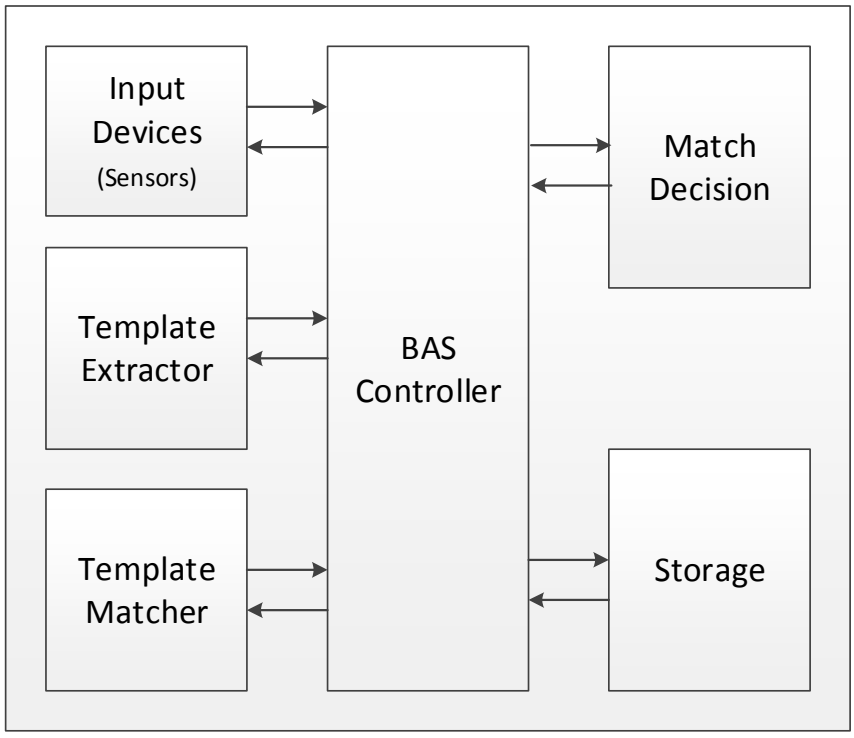

Figure 2.1: Typical Biometric Authentication System

2.4.1 Input Devices

The Input Devices module consists of sensors such as iris scanners, face camera and fingerprint sensors, which are used to capture different biometric images to form a Biometric Image Set (BIS) [24], [27]. These sensors not only acquire the biometric

images but may also consider live-ness detection, image quality assessment, image enhancement and processing [33].

2.4.2 Template Extractor

Template Extractor module uses one or more Template Extraction Algorithms (TEA) to extract salient and discriminatory features from biometric images and generate the templates out of them [29]. The generated templates together form a BTS and could be further used for matching or storage. If the BTS is used for storage, it is termed as record template RT, whereas, if it is used for matching with one or more templates in the storage, it is termed as a probe template PT where 'probe' is a term used for input data in biometrics domain.

2.4.3 Template Matcher

A Template Matcher module uses one or more Template Matching Algorithms (TMA) to perform a match between a set of probe template PT and record template RT, and generates a match score for every successful match operation [28]. An essential requirement of this module is that a successful matching could only be performed between two templates generated using the same template extraction algorithm. The underlying reason is that different vendor algorithms generate different templates for the same image based on their proprietary formats, and often two different formats are incompatible to each other for matching resulting in unsuccessful match operations [30].

2.4.4 Match Decision

A Match Decision module is responsible for making a match/no-match decision based on the value of match score generated by the template matcher module [10]. The match

score is compared with a set decision threshold value, which is based on the accuracy requirements of the system. The decision threshold value inside a Match Decision module is often kept fixed inside a typical BAS [10]. If the match score is higher than the decision threshold value, it is considered as a match otherwise a non-match.

2.4.5 Storage

The Storage module contains a biometric dataset consisting of biometric image set BIS and biometric template set BTS along with the biographic dataset BD associated with different users enrolled in the system [22], [24]. The Storage module could be a centralized server or a local machine [34], where each stored record template RT could be used by a template matcher module for further matching with an input probe template PT.

2.4.6 BAS Controller

BAS controller is the main controlling unit which manages data and process flow between different modules. Moreover, it interacts with any external system using the BAS for access control [23].

2.5 Operating Modes of BAS

Typically, a BAS operates in one of the following three operating modes: enrolment, verification and identification [22], [24]. The description of each these modes are as follows:

2.5.1 Enrolment in BAS

In enrolment mode, a storage containing biometric and biographic details of different users is created, which could be used for identification or verification purposes [22]. A series of operations involved during enrolment in a typical BAS as shown in Figure 2.2 are described as follows:

1. An enrolment request from a client application consisting of Biometric Image Set (BIS) and Biographic Dataset (BD) is handled by the BAS controller.
2. BIS is sent to the template extractor module which returns a set of templates BTS.
3. The set of templates BTS, along with the input BIS and biographic data BD is sent to the storage. The storage responds with an acknowledgement which is further returned to the client application.

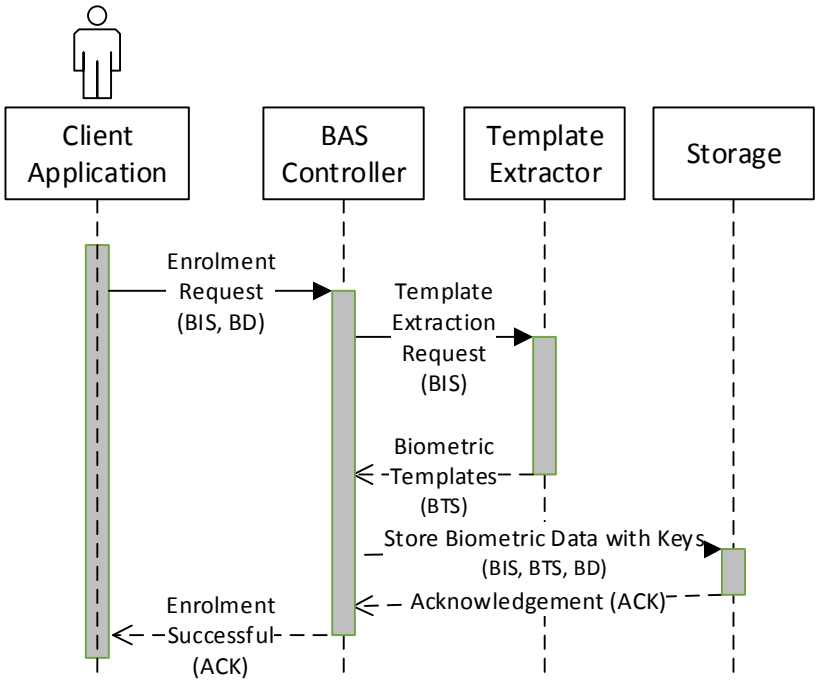

Figure 2.2: Enrolment in BAS

One of the essential aspects of enrolment, not shown in Figure 2.2, is to ensure that a particular user is not enrolled multiple times in the storage. In order to do that, the data belonging to each enrolled user is matched with the data of all other users enrolled in the storage. This process is performed to identify any duplicate records in the storage and is therefore termed as 'de-duplication search'. In large biometric systems, a process of matching a record with all the records in the storage could take a huge amount of time; therefore, a de-duplication search is often performed offline and not during the course of enrolment [9], [16] . Therefore, a user may get enrolled twice in a typical BAS which could be further removed only after a de-duplication search is performed.

2.5.2 Verification in BAS

In verification mode, a 1:1 comparison between the biometric data of two identities is performed by a BAS. The final result is in the form of a match/no-match decision which is used to determine whether the compared data belongs to the same user or two different users [24]. The detailed process flow as shown in Figure 2.3 is explained as follows:

1. A verification request handled by the BAS controller consists of a Biometric Image Set (BIS) and the claimed identity CI details of a user. CI details are often a part of the Biographic Data (BD) associated with a user.

2. BIS is sent to the template extractor module which returns a set of templates BTS. Also, CI is sent to the storage in order to retrieve the record templates RT associated with the user.

3. The Storage return the set of record templates RT associated with CI to the BAS controller. The controller further sends the RT and PT (BTS) to the matching sub-system which responds with a set of single match score MS between the two.

4. The MS is sent to the Match Decision module which compares it against the set decision threshold DT value and responds with a match/no-match decision. The decision logic corresponds to the following verification result VR:

$$VR = \begin{cases} 1, & \text{if } MS \geq DT \\ 0, & \text{if } MS < DT \end{cases}$$

where, typically a 1 corresponds to a match and 0 corresponds to a non-match

5. The VR is further returned to the client application by the BAS controller.

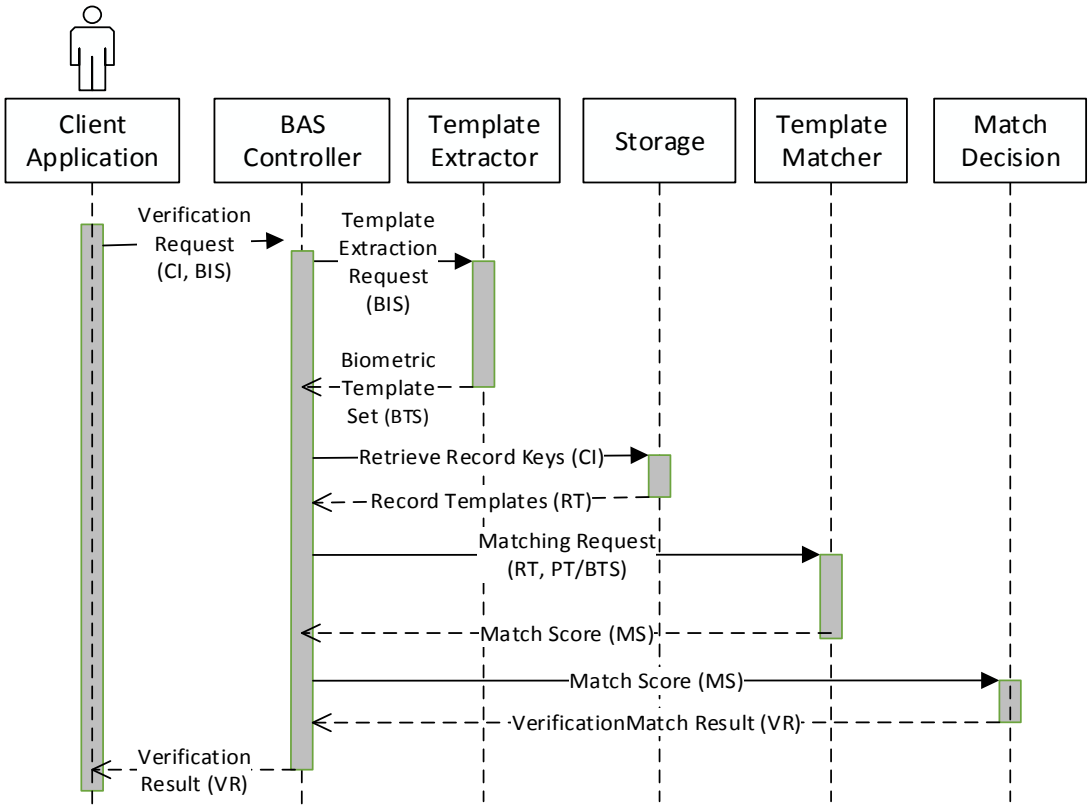

Figure 2.3: Verification in BAS

2.5.3 Identification in BAS

Identification is done in order to identify a user based on his/her biometric if he/she is already enrolled in the biometric system [30]. During identification mode, the BAS should perform a 1:N comparison and match the input biometric data with all the biometric data in storage [22]. However, in practice, the existing systems perform identification search operations over only a small subset of the total records in the storage [24]. This is done in order to ensure optimum performance of the system as matching with all the records would become similar to a de-duplication search operation and could not be performed in real-time by a typical BAS [9], [16]. The subset of records is filtered out of the total dataset in the system based on one or more biographic fields associated with different users. A typical process flow of an identification search as shown in Figure 2.4 is explained as follows:

1. An identification request containing a biometric image set BIS is handled by the BAS controller which further sends them to template extractor module to obtain the corresponding set of templates BTS.
2. Further, the set of record templates RT are retrieved from the Storage which are further sent along with the PT (BTS) to template matcher module for matching.
3. The template matcher returns a set of match scores MS for all the match operations which are further sent to the Match Decision module.
4. The Match Decision module compares each match score in MS to the pre-defined DT value and identifies whether there are one or more matching records present in the Storage.

5. In case one or more matches are found, the details of the records having match score above the DT value are sent back to the BAS controller. The BAS receives the match scores and Biographic Data (BD) associated with each matching record and sends it to the client application as the identification result.

6. On the other hand, in case when no-matching record is found, a no-match identification result typically in the form of a Negative-Acknowledgement (NACK) is sent back to the client application.

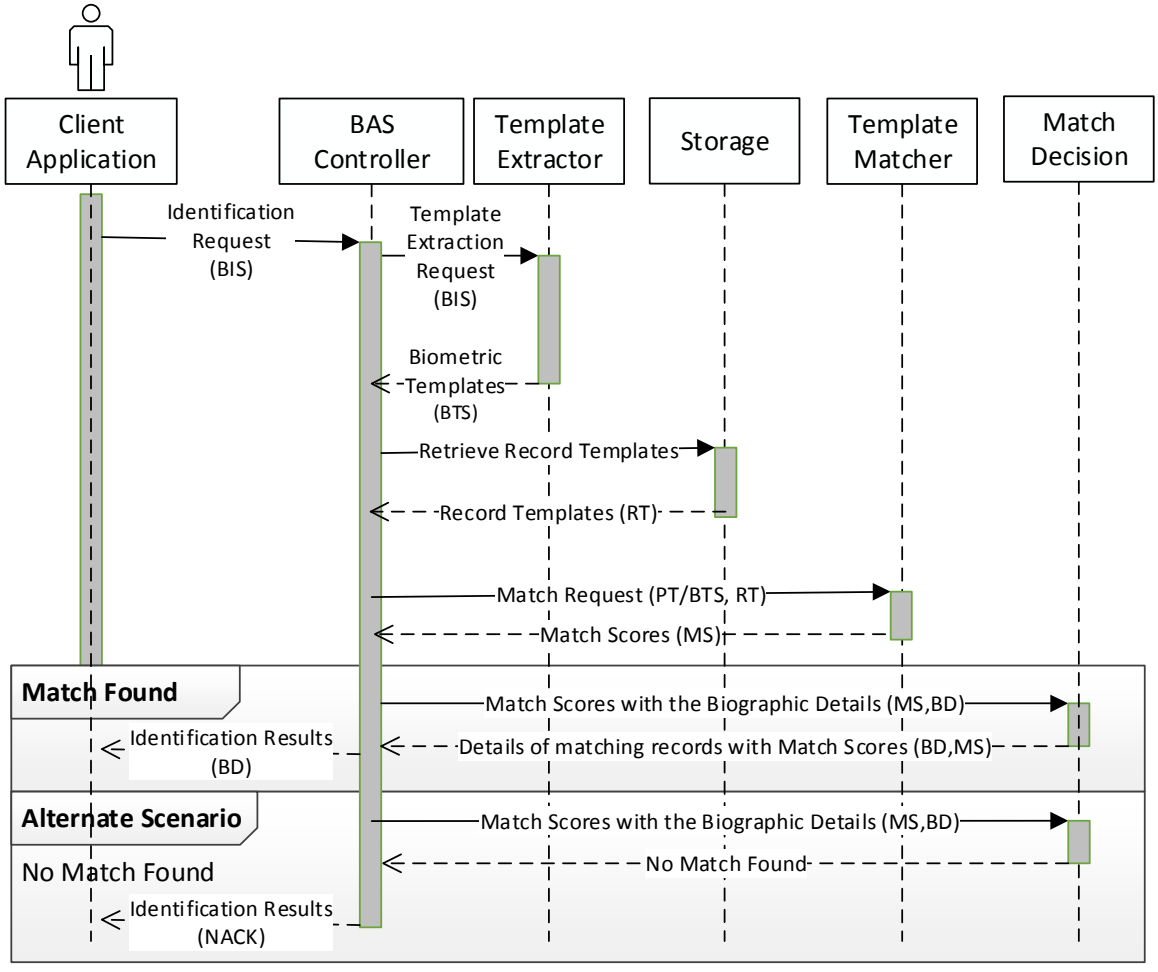

Figure 2.4: Identification in BAS

2.6 Performance Metrics in Biometrics Domain

There are some important performance metrics in biometrics domain [35] which are used throughout to evaluate the overall performance of a BAS and its individual modules such as Template Matcher and Match Decision. In biometric domain, performance is a measure of both response time and accuracy. For example, the error rate of an algorithm or a system is also a performance metric like extraction or matching response time. Some of the important biometric performance metrics [27], [36] used in this thesis are discussed as follows:

Genuine Rejection Rate (GRR): It is the fraction of the impostor match scores falling below the decision threshold DT value, meaning the impostors are rejected correctly. A higher GRR means higher accuracy of the system.

False Acceptance Rate (FAR): FAR could be defined as the fraction of impostor scores exceeding the decision threshold DT value, meaning that the impostors are accepted as genuine identities by the system. Therefore, a higher FAR means a less accurate system.

$$FAR = 1 - GRR \tag{2.1}$$

Genuine Acceptance Rate (GAR): It is the fraction of the genuine match scores exceeding the decision threshold DT value, meaning the genuine identities are recognized correctly. A higher GAR means higher accuracy of the system.

False Rejection Rate (FRR): FRR could be defined as the fraction of genuine user match scores falling below the decision threshold DT value, meaning that the genuine users are being rejected as an impostor.

$$FRR = 1 - GAR \tag{2.2}$$

Similar to FAR, a system with higher FRR means a less accurate system.

Equal Error Rate (EER): A single valued measure of a BAS performance is EER which considers both FAR and FRR together. It is defined as the point where FAR becomes equal to FRR, therefore, a lower EER indicates better performance.

2.7 Related Work

This section highlights the related work towards existing frameworks handling biometric datasets. It emphasizes on the key aspects of the frameworks specifically pertaining to biometrics domain, keeping a focus towards the major contributions made in their approach and their bottlenecks in terms of efficiently handling large biometric datasets, performing biometric operations efficiently, providing a modality-independent framework with efficient mechanisms to store and manage different types of data associated with biometrics domain, and providing mechanisms for algorithm selection in order to fulfill the requirements of different applications. Finally, the existing approaches which aim to provide performance improvement in the biometric systems are investigated in this section.

2.7.1 Approaches Limited to Single Modality

In biometrics domain, a number of frameworks [17]–[21], [37] have been proposed to handle biometric data and perform biometrics domain specific operations. These approaches are designed with a goal to store, and perform computationally expensive operations such as biometric identification and verification, over biometric datasets. However, some of these approaches [18], [19], [37] were specifically targeted towards a specific biometric modality which restricts them from being used in multi-modal systems.

For instance, Liu et al. [18] proposed a biometric authentication approach based on a novel biometric matching strategy; though the strategy is restricted to work only for fingerprints and could not be used for other modalities. Using this strategy during recognition process, the fingerprints are matched at coarse level using sparse representation technique, the difference between pores is calculated and a one-to-many pore correspondence is established between them. The evaluation of their approach shows that the underlying coarse-to-fine hierarchical strategy makes it more robust to the instability of pores and fingerprint distortions, providing a significant improvement in recognition accuracy. However, due to the specificity of the coarse-to-fine approach for fingerprint based recognition, the approach could not be extended for other modalities such as palm-print, face and iris. Similar to coarse-to-fine strategy, another approach based on downscaling the face images for performance improvement was proposed by Tao and Veldhuis [37]. Using the proposed approach, the system was able to radically reduce the number of possible classification units for detection process. It was done at two levels; first by down-scaling the face images, and second by restricting the scanning window to a fixed size, to avoid the search for images which are too small or too large. The proposed approach provided significant performance improvement and obtained an equal error rate of 2%; however, by virtue of the dependence of information fusion and other sub-systems for face biometrics, the approach could not be used for other modalities. Another similar effort towards improving the accuracy of the biometric system, but specifically restricted towards face biometrics, was made by Park & Jain [19]. The authors suggested including soft-biometric information such as gender and face-mark information along with the facial features during facial recognition. Soft-

biometric traits lack the distinctiveness and permanence to be considered alone for recognition purposes; however, when combined with hard-biometric characteristics such as facial features, they often help in improving the accuracy of the overall recognition process [38]. However, most of the soft-biometric information used in the approach such as number and location of freckles and wrinkles, moles, scars, tattoos, chipped teeth and lip creases, were specific to face recognition and could not be leveraged in the systems incorporating other modalities.

2.7.2 Scalability Limitations of Existing Approaches

Most of the existing approaches focused towards a limited set of problems and refrained themselves from solving the issues related with the handling of massive biometric datasets and are unable to provide a horizontally scalable storage for these datasets. More specifically, none of the approach focused towards the scalability issues related with storing biometric images and templates. Danese et al. [21] suggested it to store them in the RAM, Diaz-Palacios et al. [17] in RDBMS, while others [18]–[20] used the file system for the storage of the biometric data. A serious limitation of the approach provided by Danese et al. [21] is in terms of dataset size as the biometric templates are stored in the RAM. Due to the practical limitations in terms of the available sizes of the RAM today, a system incorporating millions of biometric images and templates could not leverage this approach. Further, the cost of the overall system due to the high cost of RAM compared to the disk storage could be another limiting factor and may affect the usability of the approach in a real-world system. Another system proposed by Diaz-Palacios et al. [17] stores both biometric and biographic data in a relational database. Therefore, the approach could pose severe scalability issues in a practical system storing

large amount of biometric data, or in those systems storing blobs of large size such as in case of face images or face templates. A different approach proposed by Tao and Veldhuis [37] focusing towards performance improvement of biometric recognition, paid little or no attention towards the storage of biometric datasets. They store the biometric data on a mobile device with limited memory capacity or on a PC for training and authentication with no effective means provided for data management and scalability. In general, none of the authors of the above mentioned approaches threw light to discuss over the most efficient storage mechanisms for different types of data associated with the biometrics domain.

2.7.3 Performance Bottlenecks of Existing Approaches

Most of the existing approaches did not paid attention towards the performance efficiency of their approach when applied to large-scale biometric datasets. For instance, the approach proposed by Diaz-Palacios et al. [17] store large biometric blobs in RDBMS, which is considered as a computationally expensive and inefficient task [39] and becomes increasingly inefficient as the size of an individual blob increases. Moreover, some systems require storing the biometric image along with the templates in the database, to accommodate for future biometric vendor algorithms and make their system vendor independent [9]. In those systems specifically, and those which require storing images that are multiple times larger than their respective templates, it could negatively affect the performance of the overall system. Another approach proposed by Liu et al. [18] involves multiple algorithms at different matching levels, having dissimilar performance and accuracy metrics. The approach was evaluated over a small dataset of 1480 images having 10 images for each user, still the performance of the system was found to be poor

due to multiple computationally expensive matching steps involved during recognition. Therefore, due to the low efficiency of the internal matching process, the scheme could not be employed for performing identification searches over datasets involving few million fingerprint images. Similarly, the approach provided by Park & Jain [19] was evaluated on a set of 213 input images each considered for a set of 6 to 10 facial mark types for different tests. The incorporation of facial-mark matching improved the overall recognition accuracy by 0.5% with a further improvement of 0.5% each with the addition of every biographic field. However, mark-based matching involving detection, encoding and matching of the facial marks, imposed additional overheads in terms of efficiency of the overall recognition process. The evaluation of the current approach provided a very poor facial-mark extraction rate of 15 seconds per face which could be a serious concern in a real-time online system where this delay during every input face extraction will make the overall recognition process extremely slow. Therefore, the approach could not be used practically in large biometric systems unless a significant improvement is made in terms of extraction time.

2.7.4 Support for Storing and Managing Biographic Datasets

Another serious limitation with most of the existing approaches [18], [20], [21] is that they could not provide effective storage and management of the biographic data along with the biometric data. As a result, to use them in a practical application or to obtain the benefits achieved by effectively linking the biographic data with biometric data, these systems need further improvement. For instance, the approach proposed by Tao and Veldhuis [37] provided no mechanism to store biographic data, which is very important to effectively manage the biometric data associated with different users in a practical

system. Moreover, in most of the existing approaches, there is no provision of performing indexing or querying, as the biometric blobs are stored in the file system [18]–[20] or in the system memory [21]. For instance, Peralta et al. [20] proposed a distributed framework for biometric matching over massive datasets. The approach adopts High Performance Computing (HPC) concepts to achieve high efficiency, robustness and scalability during biometric recognition process by using a cluster of servers having multiple cores. High efficiency is achieved by providing parallel search through the database, robustness is achieved through the use of multiple servers providing fault-tolerance in case of failure, and scalability is obtained by dividing the match processes across several cores and scaling the number of cores by employing more servers. Also, it does not hold the bottlenecks such as limited and costly RAM storage present in the work of Danese et al. [21] and low recognition rate provided by the approach of Liu et al. [18]. However, the authors overlooked the use of framework in a practical system which requires the biographic data of the users to be stored along with their biometric data. Moreover, due to the lack of possessing mechanisms to store biographic data, the framework does not provide any interface which could be used to query based on the user data and perform 1:N biometric identification or 1:1 verification between user data effectively. On the contrary, the approach provided by Park and Jain [19] stores biographic data and could perform biographic matching as well; however, it could only store and match the binary fields such as gender and has no mechanism for handling the non-binary fields such as nationality, address, first name, last name which are equally important for a real-world biometric system.

2.7.5 Uniform Interface Support for Biometric Systems

Most of the existing architectures succumb a serious bottleneck of not being able to provide a uniform interface to access the underlying storage in a biometric system. The approach provided by Diaz-Palacios et al. [17] addresses this issue up to some extent by providing an SQL layer; however, as discussed their approach is not very efficient for large-scale biometric systems containing massive datasets of biometric images and templates.

2.7.6 Mechanisms for Biometric Algorithm Selection

A common limitation with all of the discussed approaches is that they did not provide any mechanism for selecting between different biometric algorithms, which is often necessary for those applications which pose strict restrictions in terms of template extraction and matching performance, template size, and accuracy metrics such as FAR and FRR requirements [30]. The existing approaches are bound to either one or more algorithms integrated with their systems, and could not provide mechanism to choose a particular algorithm over other based on different application requirements such as those mentioned above.

2.7.7 Approaches for Performance Improvement

In the literature, there are some existing approaches [40]–[43] which could be used as a part of a biometric system in order to reduce the overall search space during identification search operations. These approaches provide methodologies towards improving the performance of the biometric systems. A majority of these approaches [40]–[42] are targeted towards iris biometric systems, since iris is considered to be one of the most accurate biometric modality [42] and as a result is used in some of the large scale

biometric systems [8], [14]. The approach proposed by Rathgeb and Uhl [40] adopted biometric hashing technique where low-dimensional hash values are generated for different iris images in the database, which are further used as keys for reducing the overall search space during an identification search operation. Another similar approach proposed by Mehrotra et al. [41] for iris based identification systems, works on energy-histogram method in order to provide search space reduction. In this method, the feature vector of each iris image is divided into different energy values from 10 different sub-bands. The histogram generated from each sub-band is further classified into bins to form logical groups of the iris image strips having similar energy values. The bin number for each images are used as a key for reducing the overall search space during identification search operations. Proenca [42] proposed a different approach aiming towards low quality iris images where the feature space of each image is decomposed into multiple scales and are placed in an n-ary tree based on their most reliable components. During identification search operations, each probe image is also decomposed into multiple scales and the distance of each centroid is used to determine the paths in the tree to find the identity of interest. All of the above discussed approaches provide performance improvements at different levels during identification search operations; however, a common limitation with all of them is that they operate at the feature–level and hence, could not be applied to modalities other than iris and are unusable for multi-modal biometric systems. In contrast to the above approaches, a different multi-modal approach using feature-level fusion and Kd-Tree for reducing the data retrieval time during identification search operations is provided by Jayaraman et al. [43]. The feature-level fusion technique used in this approach performs dimension-reduction and selects only

top-10 eigen values out of the larger dimension space of 64 or 88 dimensions for different modalities. Since, the quality of a biometric image highly depends on the acquisition conditions at the time when biometric image was captured; this technique may not do well with poor quality images as the accuracy of the feature space becomes an issue. Moreover, all of the above mentioned approaches are solely aimed towards performance improvement in biometric systems and did not focus on providing the scalability solution for storing and managing large biometric datasets.

2.8 Summary

In this chapter, a description of biometric authentication technology and the underlying concepts are presented. A typical biometric authentication system, its various sub-systems, operating modes and various performance metrics associated with it were described as a next step. Further, different existing approaches were discussed to highlight their individual contributions in biometrics domain and were analyzed to identify the bottlenecks in each one of them towards handling massive biometric datasets and addressing its associated issues. Each of the discussed approaches focused only to a subset of problems in biometrics domain and none of them adopted a holistic approach towards solving the issues related to handling massive biometric datasets and providing an optimum storage for these datasets. To the best of my knowledge, the kind of work which provides a horizontally scalable storage and simultaneously addresses the performance issues related to large-scale biometric systems has not been carried out till today. Therefore, there is an inevitable need to design a multi-modal biometric framework for efficiently performing biometric operations and simultaneously providing an effective storage for large-scale biometric systems.

Chapter 3

3 Data Storage Technologies

This chapter discusses about the various approaches towards data storage and database management systems, in order to identify the suitable data storage for handling massive datasets in biometrics domain. These datasets consists of both biographic and biometric data with different storage requirements. Therefore, it is important to study the different data storage options available today before selecting the optimum storage for each type of data.

While discussing databases, Relational Database Management Systems (RDBMS) are discussed first since they have been playing a dominant role in the industry during the past few decades [44]. However, due to the recent needs of scalable data storage and processing, a new category of data stores known as NoSQL (Not only SQL) came into existence and is discussed further.

3.1 Relational Database Management Systems (RDBMS)

RDBMS had been a preferred choice for database management and played a dominant role among other storage technologies such as object databases and XML databases during last few decades [44]. The RDBMS data model consists of a collection of tables and their relationships, where each table contains several rows/records and columns. However, the number of columns and the data type each column can hold becomes fixed once it is defined. Therefore, RDBMS are said to have fixed schema as any addition of a column after a table is designed, requires redesigning the table. However, RDBMS

provide ACID (Atomicity, Consistency, Isolation and Durability) compliance and transactional integrity with concurrency control [45]. They also offer flexible indexing and querying capabilities. Structured Query Language (SQL) provides a standard interface to communicate and perform complex querying through different RDBMS. Moreover, RDBMS offer powerful security features such as data encryption, authentication, authorization and auditing.

RDBMS provides a rich set of features and tools which together makes it an effective data management technology useful for different application scenarios. However, it possesses certain set of challenges when dealing with massive datasets which needs careful attention and are discussed further.

3.1.1 Inefficiencies of RDBMS

RDBMS evolved at the time of limited computing capabilities and limited data processing needs. Today, with the growth in the number of enrolments and an increased use of multiple modalities in biometric systems, large data processing and storage has become an essential requirement for these systems. However, RDBMSs possess certain set of challenges when dealing with massive biometric datasets in the following ways:

1. Biometric datasets are massively large in the existing large scale BAS and are growing at a very fast pace requiring scalable storage beyond the capabilities of RDBMS [15]. RDBMS provides vertical scalability which has obvious limitations in scaling up to the capacity of the largest servers available today. This is especially true for the case of biometric systems having large number of enrolments and multiple modalities [8], [9].

2. Another major limitation with RDBMS arises in terms of handling massively concurrent and fast reads and writes. In biometrics domain, where each biometric template matching process is inherently parallel, the RDBMS could possess bottlenecks in terms of providing data access for different processing needs.
3. In RDBMS, impedance mismatch requiring object-relational mapping has always been a complicated and performance inefficient process [46]. It becomes even more severe while dealing with large blobs of biometric images and templates.

The different types of data in biometrics domain comprises of biographic and biometric datasets. The biographic data contains details about different users such as name, address, gender and personal identification numbers. This type of data is structured and often requires powerful indexing and querying based on the different fields of biographic data. Moreover, this data is often quite sensitive as it contains personal information about different users, requiring secure mechanisms for data storage. Therefore, analyzing the benefits of RDBMS in terms of providing flexible indexing and querying capabilities with a uniform access interface, along with powerful security mechanisms, they could be an optimum choice for the storage of biographic data associated with different users. Therefore, the Hybrid Data Storage Framework proposed in this research uses RDBMS for storing biographic datasets, association of different biometric images with the biographic data, association between different biometric images and their templates, modality and sub-modality details of biometric images and template, and the keys associated with each biometric image and template. Several implementations of RDBMS are available such as Microsoft SQL server, Oracle database, Oracle MySQL database

and IBM DB2; however, in this research the open-source Oracle MySQL database has been chosen for the implementation of RDBMS storage in HDSF.

However, considering the efficiencies of RDBMS in terms of handling massive biometric datasets, there is a need to choose a different storage mechanism for storing biometric templates and images. The limitations of RDBMS in terms of handling massive datasets in biometrics as well as other domains, led to the development of another class of data stores capable of handling the requirements posed by massive datasets. This newer class of data stores is termed as NoSQL (Not only SQL) and is discussed in the next section.

3.2 NoSQL Data Stores

NoSQL is used as an umbrella term for the class of data stores that do not exactly follow the traditional RDBMS concepts such as ACID compliance, SQL style querying, and fixed schema. On the other hand, NoSQL data stores offer flexible schema or are sometimes completely schema-free and are designed to handle a wider variety of data than just tables as it was with RDBMS [47]–[49].

NoSQL databases could be broadly classified into four categories: Column family stores, Document stores, Graph databases and Key-value stores [48], [50], [51]. The different types of data stores vary widely in terms of their capabilities and the features offered by them. Therefore, in order to use different NoSQL data stores, it is important to identify their capabilities and how they differ from the traditional RDBMS systems.

3.2.1 Column-family Stores

Column-family stores are derived from Google Bigtable [52], where the data is stored in column-oriented way. The dataset consists of several rows each of which is addressed by a unique row-key, also known as primary-key. Each row is composed of a set of column-family and the data pertaining to a row-key is stored together as shown in Figure 3.1. However, each column-family further acts as a key for one or more columns it holds, where each column is comprised of a name-value pair. Further, a column does not exist if it contains a null value which improves storage requirements.

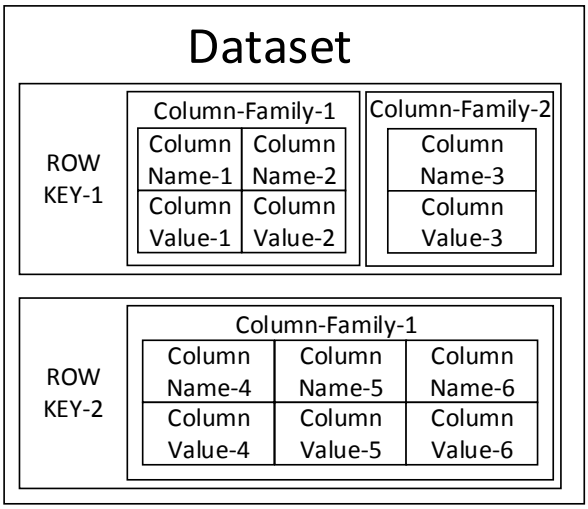

Figure 3.1: Column-family Store Data Model

Column-family stores provide high-horizontal scalability; however, the indexing and querying capabilities are limited at the column-family and column level. Moreover, any logic requiring relations needs to be implemented in the client application. In contrast to RDBMS, column-family data stores require storing the same data multiple times for efficient querying. This poses another limitation towards data integrity as any update in a data field may require several data points to be updated simultaneously.

In view of the strengths and inefficiencies of the column-family stores, it is evident that they are not suitable for storing the biographic datasets as they cannot provide support for relations and possess serious limitations towards data integrity. Moreover, their data model is overly complex for storing biometric datasets which consists of blobs of images and templates along with their associated keys.

3.2.2 Document Stores

Document stores provide data storage in the form of documents, where each document could be accessed by a unique document id as shown in Figure 3.2. Most of the document data stores represent documents using the JSON (JavaScript Object Notation) [53], or some format derived from JSON. Document stores are suitable for applications where the input data could be represented in a document format as mentioned above. However, a document could contain complex data structures such as nested objects, and does not require adherence to a fixed schema. Moreover, document stores provide the capability of indexing documents based on the primary-key as well as on the attributes of a document.

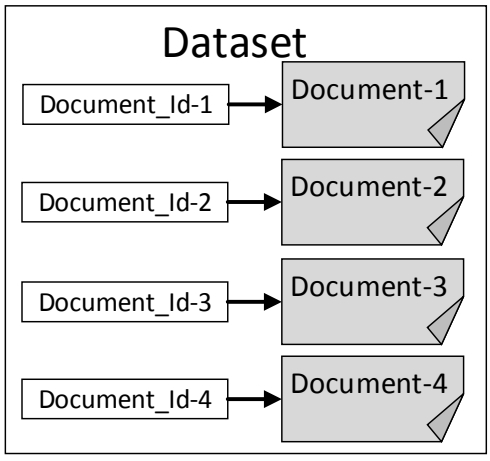

Figure 3.2: Document Store Data Model

Similar to column-family stores, document stores too are not suitable for storing the biographic datasets as they cannot provide support for relations and lack data integrity. Moreover, their data model is only suitable for storing data types similar to documents, which again becomes excessively complex for storing biometric datasets which consists of blobs of images and templates along with their associated keys.

3.2.3 Graph Databases

Graph databases originated from graph theory, use graphs as their data model. A graph is a mathematical concept used to represent a set of objects, known as vertices or nodes, and the links (or edges) that interconnect these objects. Graph databases possess a completely different data model than column-family and document stores, where the nodes and edges have individual properties comprising of key-value pairs for data storage as shown in Figure 3.3. Graph databases are specialized in handling highly interconnected data, and, therefore, are very efficient in traversing through relationships between different entities. Traversing involves visiting nodes in a graph until the required relationship between desired nodes is established.

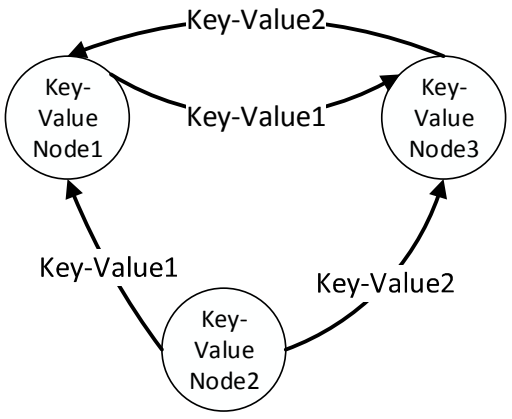

Figure 3.3: Graph Database Data Model

Other than the purpose of storing relations, graph databases are not as efficient as other NoSQL data stores in terms of horizontal scalability and storing massive distributed datasets. Therefore, they address only a very small subset of requirements in biometrics domain, and are unsuitable to be considered for storing either biographic or biometric datasets in HDSF data storage.

3.2.4 Key-value Stores

Key-value stores hold a very simple data model based on key-value pairs, resembling a map or a dictionary as shown in Figure 3.4. The key uniquely identifies the value and is used to store and retrieve the value into and out of the system. The value is opaque to the data store, and could be used to store any arbitrary data including an integer, a string, an array or an object, providing a schema-less data model. Key-value stores are very efficient in storing huge distributed data. However, they cannot handle data level querying and indexing since the values are opaque to the data store. Moreover, they cannot implement relations, and any functionality requiring relations needs to be handled by the client application interacting with the key-value store.

Key_1	Value_1
Key_2	Value_2
Key_3	Value_1
Key_4	Value_3
Key_5	Value_2
Key_6	Value_1
Key_7	Value_4
Key_8	Value_3

Figure 3.4: Key-value Store Data Model

Considering the strengths and inefficiencies of the key-value stores, it is evident that they are not suitable for storing the biographic datasets as they cannot provide data level querying and indexing capabilities required by biographic datasets. However, they could be extremely suitable for storing biometric images and templates which are often stored as biometric blobs and are inherently unstructured. Moreover, since key-value stores are very efficient in terms of storing huge distributed data, they could scale up to the needs of large biometric systems containing millions of enrolments together with comprising of multi-modal biometric datasets. Therefore, the Hybrid Data Storage Framework proposed in this research uses Key-value type of storage for storing biometric datasets including the biometric images and templates and their associated keys. Several implementations of Key-value stores are available such as Memcached [54], Amazon DynamoDB [55], Azure Table Storage [56], Redis [57] and Riak [58]; however, in this research the open-source Redis Key-value store is chosen for the implementation of NoSQL Distributed Data Storage in HDSF. The reason behind choosing Redis among other data stores are the following:

- Among all of the available Key-value stores, some of them such as Azure Table storage and DynamoDB have a closed source license with a pricing associated with their use. Therefore, a freely available open-source Key-value store such as Redis, Riak or Memcached is preferred for the evaluation in this research.
- Redis allows sending multiple commands in a single write operation through pipelining, which helps in reducing the overall response time of the system. Redis is a TCP server and uses a client-server model, where the server processes the commands sent by the client as a query, and sends back the responses [57]. In this

research, sending multiple commands is required during the evaluation of the proposed framework while performing multiple match operations during de-duplication, identification and verification processes.

- Redis supports horizontal partitioning of data across multiple servers so that each server only contains a subset of the total data [57]. Moreover, the partitioning scales the computational power to multiple cores across different servers. This feature is used in HDSF where the biometric data is distributed across different servers, where each server processes its own set of data using independent match engines as discussed in the following chapter.
- Redis provides atomic transactions where either all or none of the commands are processed. All the commands in a transaction are serialized and executed sequentially and a request from another client is not processed in between the transaction, guaranteeing the execution of commands as a single isolated operation. During the evaluation in this research, since there is only one client application, the master ensures that a new transaction is not issued to the servers until the previous one completes.
- Redis provides a system termed as 'Redis Sentinel' which is designed to manage Redis instances. It performs the following tasks such as: monitoring, for constantly checking whether the master and slave servers are working as expected; notification, in order to notify the client application via an API about an error in one or more Redis instances; and automatic failover, to promote one of the slave as master in case the master is not working and configure other slaves to use the new master server. 'Redis Sentinel' is a distributed system where each

server runs its own sentinel process and uses gossip protocol [57] for communication between different processes. Although, the evaluation in this research does not uses 'Redis Sentinel', it could be a useful asset for an application managing large number of Redis servers.

It is important to note here that the choice of using Redis is made specifically for the evaluation in this research. Therefore, any other Key-value store could be chosen in place of Redis depending upon a particular application requirement, without affecting the overall performance improvement obtained due to the index creation, key based filtering and matching processes proposed in the following chapter in this research.

3.3 Summary

In this chapter, data storage technologies were discussed in detail, starting from the much established and prominent RDBMS to the more recent NoSQL data stores. On the one hand, RDBMS pose limitations when dealing with massive datasets; whereas on the other hand they provide flexible indexing and querying capabilities along with data integrity and powerful security features required for biographic datasets. In contrast to RDBMS, most of the NoSQL data stores, except Graph databases, provide high horizontal scalability but less powerful indexing, querying and security features. However, the key-value stores provide the most suitable storage for storing biometric images and templates, above all other NoSQL data stores. Moreover, considering the fact that none of the RDBMS or Key-value NoSQL data store alone could cater to the different data needs posed by biometric systems, there is an inevitable need to adopt a hybrid approach using both, in order to store the different variety of data in biometrics domain.

Chapter 4

4 Hybrid Data Storage Framework

A Hybrid Data Storage Framework (HDSF) is presented in this chapter, to address the issues pertaining to scalability and performance in the existing BASs, while dealing with massive biometric datasets. HDSF aims to provide enhancements over traditional BAS, by proposing the following new approaches:

- A hybrid, horizontally scalable, data storage approach for biometric systems, to support scalability requirements of large-scale identity matching systems storing large biometric datasets.
- A set of four new processes to enhance performance at multiple levels over a traditional BAS; those are **(a)** Index Profile Creation and Data Storage, **(b)** Biographic Match Score based Key Filtering, **(c)** Multi-modal Biometric Index based Key Filtering, and **(d)** Key based Biometric Matching. These processes improve the performance of identification and de-duplication search operations. The details regarding each of these processes are provided in section 4.2.2.

In addition to providing the scalability and performance related improvements, HDSF also provides the following enhancements over traditional BAS:

- By providing a new approach for on-the-fly selection of different biometric template extraction algorithms to serve the requirements of different applications in terms of accuracy and efficiency as discussed in section 4.2.1. Further, an

adaptive multi-modal matching approach is proposed for each individual match operation in section 4.3.3.

- By providing a biometric modality independent interoperable framework, providing no limitations towards inclusion of any number and type of biometric modalities, which improves the overall usability of the framework.

- By providing the capability to store and efficiently manage the biographic data associated to different users, with the use of RDBMS which provides indexing and querying over the biographic datasets. The relational database schema of the existing biometric systems which are based on RDBMS, could be easily migrated to HDSF, where only the biometric data is stored separately in NoSQL storage while keeping rest of the schema unchanged.

- By providing an Application Programming Interface to access the internal functionalities offered by HDSF, while abstracting the details of the different storage mechanisms used for biographic and biometric data. Further, this API is exposed as a service in order to enable access through different applications and devices.

The proposed Hybrid Data Storage Framework (HDSF) consists of a layered architecture comprising of the following layers as shown in Figure 4.1: Web-Service based Application Programming Interface (API) layer, Biometric Biographic Management (BBM) layer and Storage and Processing layer. The details of each of these layers are explained in the subsequent sections.

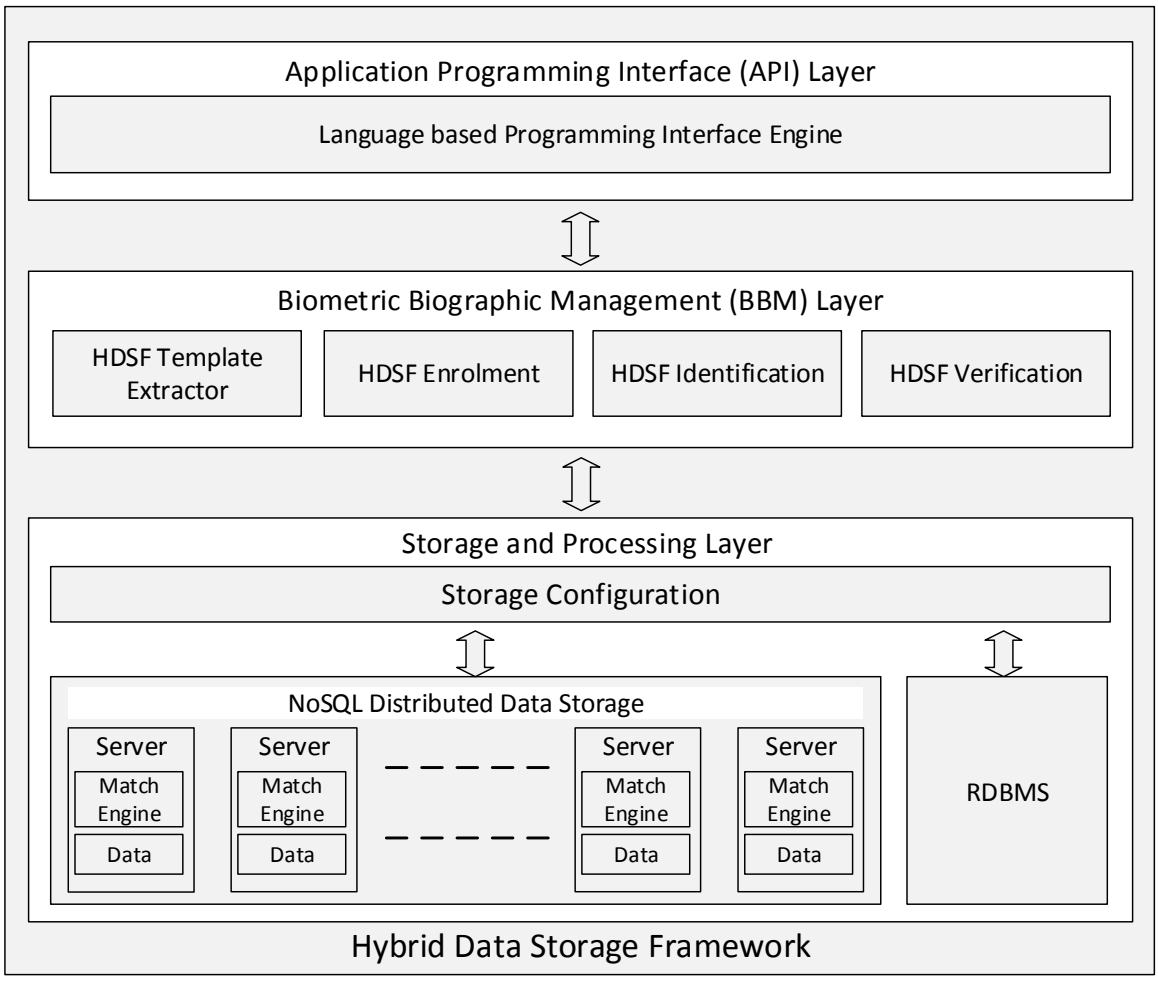

Figure 4.1: Hybrid Data Storage Framework

4.1 Application Programming Interface (API) Layer

The API layer is offered as a web service and provides an interface to HDSF. Any external system or client application interacting with HDSF will communicate through the API layer. It exposes the internal functionalities of HDSF by using a Language based Programming Interface Engine**.** This engine is provided in order to support easy integration with the applications using a programming language interface. Moreover, the interface being offered as a service enables access to HDSF through different devices and

platforms. The API layer provides a uniform access interface by abstracting the internal access details for different biographic and biometric data storage.

4.2 Biometric and Biographic Management (BBM) Layer

This layer comprises of modules handling specific functionalities related to biometrics domain. It consists of the following modules: HDSF Template Extractor, HDSF Identification, HDSF Enrolment and HDSF Verification. The functionalities for each of these modules are explained as follows:

4.2.1 HDSF Template Extractor

The HDSF Template Extractor module is used by Enrolment, Identification and Verification modules inside Biometric Biographic Management layer. A new approach is proposed and is used by HDSF Template Extractor module in order to achieve the goal of providing on-the-fly algorithm selection during the template extraction process in HDSF. In this approach, HDSF Template Extractor module maintains information about different algorithms such as algorithm name, supported modality, average template size, template extraction and matching time, in an algorithm configuration file as shown in Figure 4.2. The algorithm configuration file is created manually and the information about an algorithm is updated in the file, whenever a new algorithm is integrated to HDSF. The information about different algorithms contained in the file is provided to a client application, upon request through the *GetBiometricAlgorithmDetails* API function mentioned in section 5.2.1. The client application could then select an appropriate template extraction algorithm using a different *SetBiometricAlgorithm* API function mentioned in section 5.2.1. The selection of an algorithm could be based on different

application specific criteria such as template extraction time, template size, template matching time and accuracy. For example, an application having strict requirements regarding maximum template size could choose a particular algorithm over the other which meets the template size criteria. The algorithm selection API function ensures that only one algorithm belonging to a particular modality is set as a default algorithm to avoid conflicts during template extraction. For example, in Figure 4.2 both the algorithms 'VeriEye' and 'Mirlin' belong to iris; however, only one of them could be set as default at a particular point of time. The algorithm set as the default is further used for all the template extraction processes related to the specific modality.

```
<biometricalgorithms>
  <algorithm id="1">
    <name>VeriFace</name>
    <modality>Face</modality>
    <averagetemplatesize>36 Kilobytes</averagetemplatesize>
    <averageextractiontime>270 millisecond</averageextractiontime>
    <averagematchingtime>4 millisecond</averagematchingtime>
    <vendorname>Neurotechnology</vendorname>
    <default>Yes</default>
  </algorithm>
  <algorithm id="2">
    <name>VeriEye</name>
    <modality>Iris</modality>
    <averagetemplatesize>3 Kilobytes</averagetemplatesize>
    <averageextractiontime>156 millisecond</averageextractiontime>
    <averagematchingtime>1 millisecond</averagematchingtime>
    <vendorname>Neurotechnology</vendorname>
    <default>Yes</default>
  </algorithm>
  <algorithm id="3">
    <name>Mirlin</name>
    <modality>Iris</modality>
    <averagetemplatesize>15 Kilobytes</averagetemplatesize>
    <averageextractiontime>235 millisecond</averageextractiontime>
    <averagematchingtime>3 millisecond</averagematchingtime>
    <vendorname>SmartSensors</vendorname>
    <default>No</default>
  </algorithm>
```

Figure 4.2: Algorithm Configuration File

The HDSF Template Extractor module takes biometric raw images BIS as input and provides the corresponding biometric template set BTS as output to the different modules inside HDSF as shown in Figure 4.3.

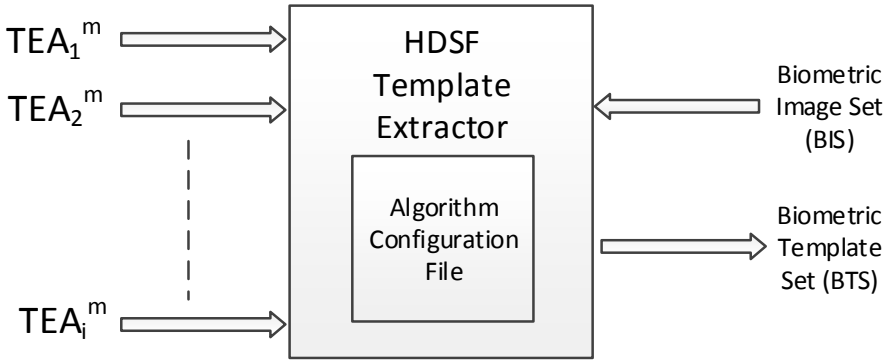

Figure 4.3: Template Extraction Algorithm Selection

Once an algorithm is set as a default for a particular biometric modality m, the sub-modality s associated with each image BIS_i^s is used to select a particular template extraction algorithm TEA_i^m such that sub-modality s belongs to modality m. The subscript i in BIS_i^s denotes the specific biometric image and in TEA_i^m represents the algorithm ID in the set of algorithms TEA. The algorithm for the proposed approach is given as follows:

for each (BIS_i^s in BIS)

% Use selected algorithm to Extract Template

$BTS_i^s = TEA_i^m(BIS_i^s)$ such that s belongs to modality m and TEA_i^m.default = Yes

end

where m represents modality, s represents sub-modality and

i denotes the count of a specific image, template or algorithm ID in their sets

Overall, the HDSF Template Extractor module is an improvement over the Template Extractor in a traditional BAS, and provides benefits to a large number of different

application scenarios which often pose strict restrictions in terms of template size, template extraction and template matching efficiency [30]. As different biometric extraction algorithms may adapt totally different approaches for feature extraction, they could result in creating dissimilar biometric templates for the same raw biometric image. Therefore, by accessing the algorithm details from the configuration file, a particular algorithm may be chosen over other by a client application, considering its average template size, template extraction and matching efficiency.

4.2.2 HDSF Enrolment

An enrolment process involves storing the biographic and biometric data of a user in the storage. In HDSF, the biographic data of the user is stored in RDBMS and the biometric data is stored in the NoSQL storage, where a set of keys is used to link the two types of data together. A storage configuration module handles different enrolment, identification and verification requests sent by the modules in biometric biographic management layer, and provides the required data storage and processing by using RDBMS and NoSQL storage in order to serve those requests. For every user, the biometric data consists of a set of biometric images BIS and templates BTS related to different sub-modalities. Each image in BIS and each template in BTS is associated with a unique key generated by RDBMS. These keys along with their associated images and templates are stored in the NoSQL storage. Also, these set of keys along with the match-score index values associated with different templates for a user are stored in the RDBMS, as explained further in this section. The enrolment in HDSF provides performance improvement over enrolment in the traditional BAS, by proposing the following four processes:

- Index Profile Creation and Data Storage,

- Biographic Match Score based Key Filtering,
- Multi-modal Biometric Index based Key Filtering, and
- Key based Biometric Matching.

The first process of *Index Profile Creation and Data Storage* is a fundamental process during enrolment in HDSF, which is followed by the next three processes for performance improvement during de-duplication search operation. However, the *Index Profile Creation and Data Storage* process is preceded by another process known as *Reference Image Enrolment* process which is performed only once during the initial use of HDSF. Therefore, the *Reference Image Enrolment* process is not provided in the list of proposed processes and is discussed first followed by the discussion of *Index Profile Creation and Data Storage* process. Further, the last three key filtering and matching processes are discussed in detail which provide online de-duplication search during enrolment.

Reference Image Enrolment Process: A reference image enrolment process is carried out to enroll a set of good quality biometric reference images during the initial use of HDSF. Once a set of reference images is stored in the system, they are kept fixed for all the future operations in HDSF. The steps for the process as shown in Figure 4.4 are as follows:

1. A set of biometric reference images RFI along with its biographic data BD are sent by a client application using an enroll reference image request to the HDSF Enrolment module through the API.
2. The HDSF Enrolment module uses HDSF Template Extractor to generate a set of biometric reference templates RF corresponding to the images in RFI. The set of

templates RF consists of templates for each sub-modality RF_i^s where s denotes sub-modality and i denote the template count in the set of reference templates RF.

3. The reference templates RF along with images RFI and biographic data BD are sent to the storage configuration module. The storage configuration module sends the RF, RFI and BD to RDBMS and obtains a set of keys KRFI and KRF corresponding to images RFI and template RF, respectively. The images RFI and templates RF are not stored in RDBMS, but are used by the RDBMS to generate the unique keys in the sets KRFI and KRF.

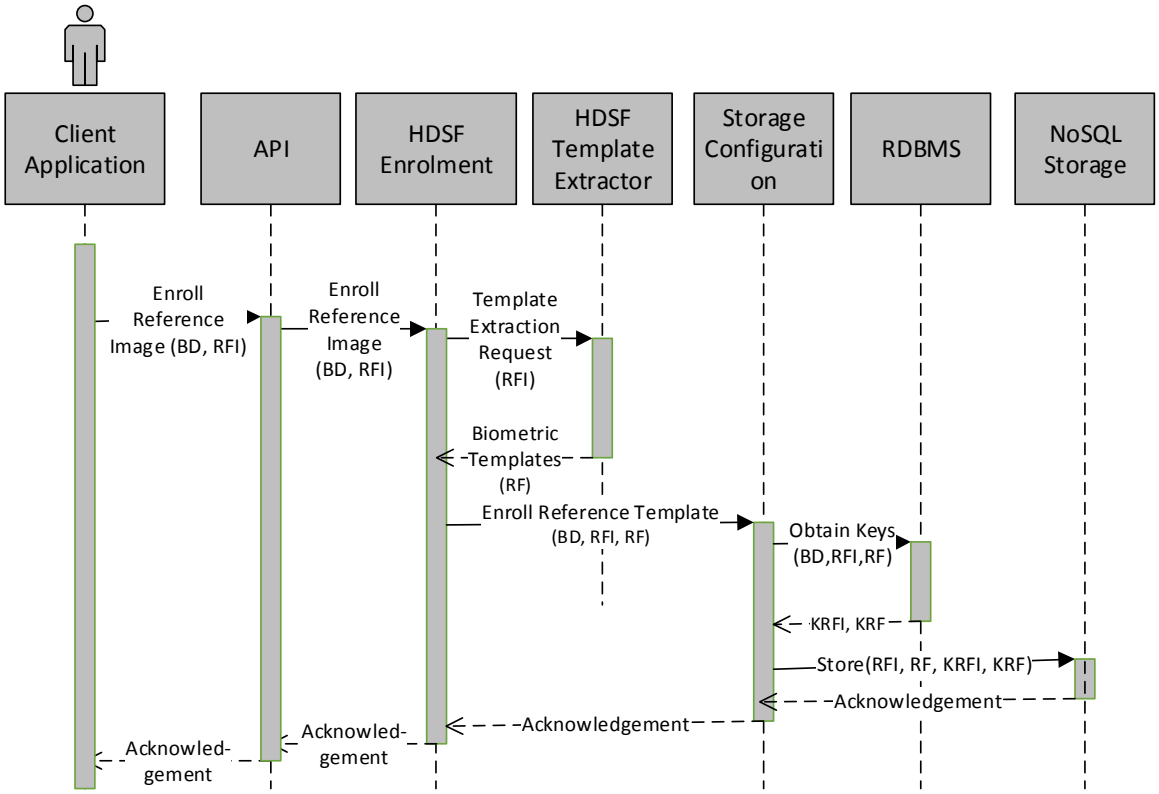

Figure 4.4: Reference Image Enrolment in HDSF

4. Finally, the image RFI and template RF are stored in the NoSQL storage along with the corresponding keys KRFI and KRF. The NoSQL storage stores the

reference images and template KRFI and KRF in each of its servers to perform parallel match operation during the future use of the system. Once the images and the templates along with their keys are stored in NoSQL storage, it responds with an acknowledgement which is further sent back to the HDSF Enrolment module and finally to the client application through the API.

Index Profile Creation and Data Storage: Once a set of reference images are enrolled in HDSF, they are kept fixed for all the future operations in HDSF. In case one or more reference images are changed, the process of *Index Profile Creation and Data* Storage has to be redone before using the HDSF for further enrolment, identification or verification processes. The steps for the process as shown in Figure 4.5 are given as follows:

1. An index profile creation and data storage request sent by the client application is handled by the enrolment module. The request consists of a set of biometric images BIS along with the associated biographic data BD.
2. The HDSF Enrolment module uses HDSF Template Extractor to generate a set of biometric templates BTS corresponding to the images in BIS.
3. The set of templates BTS along with BIS and BD are sent to the storage configuration module as an index profile creation and data storage request.
4. The storage configuration module sends the set of probe templates PT (in biometrics domain probe word is used for input data) to the NoSQL storage which matches them with the reference templates RF stored in the NoSQL servers and returns a set of match-score index MSI. The Match Engine inside each NoSQL storage server is used to match each probe template PT_i^s (s = sub-modality, i =

template count in the set BTS) with the corresponding Reference Template $RF_i^s \in RF$ to generate a Match Score Index MSI_i^s (s = sub-modality, i = match score count in the set MSI). Each MSI_i^s is further used as an index value for the particular template PT_i^s. This process is repeated for each template $PT_i^s \in PT$ which generates a corresponding Match Score Index for each sub-modality template forming a set such that $MSI_i^s \in MSI$.

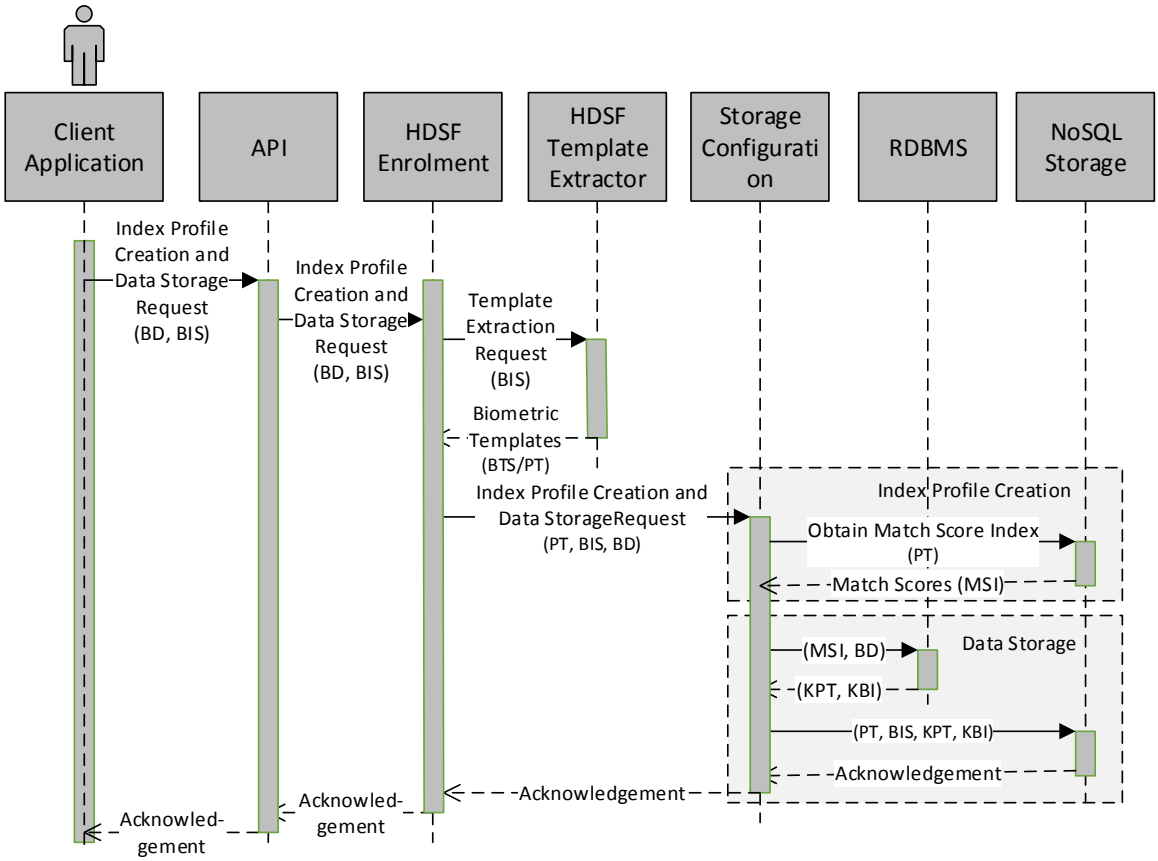

Figure 4.5: Index Profile Creation and Data Storage Process

5. The MSI_i^s along with the biographic data BD belonging to the user, is stored in RDBMS. RDBMS generates a set of unique keys KPT corresponding to the set of

probe templates PT, and another set of unique keys KBI for the set of images BIS, which are returned to the storage configuration module.

6. As a next step, the probe templates PT and the images BIS along with the keys KPT and KBI are stored in the NoSQL storage. Internally, the Storage Configuration module uses the match score index MSI_i^s value corresponding to a probe template PT_i^s, to determine the storage server SJ in NoSQL storage which stores PT_i^s and the associated image BIS_i^s. The storage server in the NoSQL storage is determined using the following equation,

$$J = Modulo(MSI_i^s / N) + 1, \text{ where N=Total number of Servers} \quad (4.1)$$

In order to explain the use of match-score index MSI_i^s value for storing PT_i^s, let us consider an example where MSI_i^s comes out to be 75 on a scale of 0-100. Further assuming that the system has 4 servers and this number is not changed during run-time inside the NoSQL storage, the different servers are termed as S1, S2, S3, and S4 as shown in Figure 4.6, and the storage is calculated as: $J = Modulo((MSI_i^s = 75) / (Server\ Count=4)) + 1 = 4$. Therefore, the server holding the template will be S4, as J = 4.

In case the value of J is a decimal value, the value is rounded off to the next integer in order to determine the storage server. Moreover, if the total number of servers is changed, the new value of N is used depending upon the server count and the whole dataset needs to be re-partitioned among the servers. All the templates in PT along with its keys KPT, and the images in BIS along with the keys KBI are stored in the server determined by using the MSI value. The NoSQL storage responds with an acknowledgement which is sent by the storage

configuration module to the HDSF Enrolment module and finally to the client application.

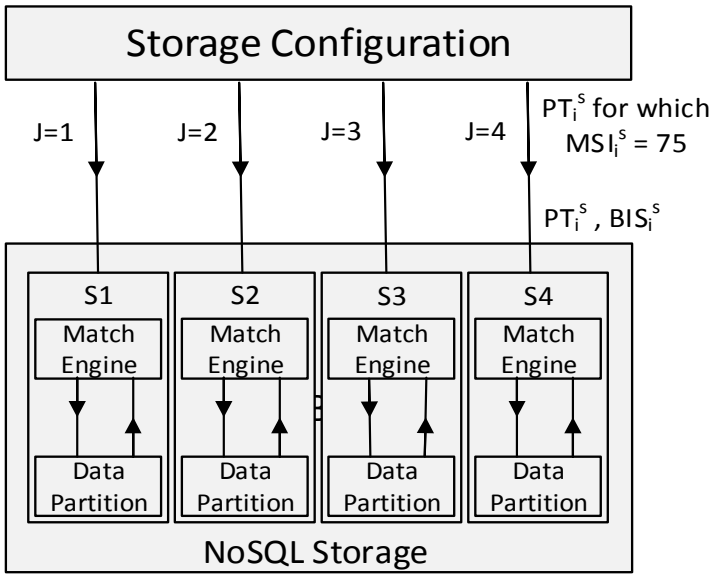

Figure 4.6: Determining Storage Server based on Match-Score Index Value

The algorithm for the Index Profile creation and storage process is given as follows:

% Index Profile Creation

for each $PT_i^s \in PT$ % Create MSI for the set of probe templates

MSI_i^s = Match each PT_i^s with RF_i^s % Obtain match score index

% Create a set of match score indexes

$MSI = \bigcup_{i=1}^{n} MSI_i^s$ where n = total number of match score index values

end

% Data Storage

Store MSI and BD in RDBMS and Obtain keys KPT and KBI from RDBMS

Determine storage server SJ where J = Modulo (MSI_i^s / Server Count) + 1

for each $PT_i^s \in PT$

Store PT_i^s, KPT, BIS_i^s and KBI in storage server SJ

end

where s represents sub-modality and

i corresponds to the particular template or a match score index value count

As shown in the algorithm, the index profile creation involves generating a set of match score indexes MSI by matching each probe template PT_i^s (s = sub-modality, i = template count in the set BTS) with the corresponding Reference Template $RF_i^s \in RF$. The set of match score indexes MSI along with the biographic data BD is stored in the RDBMS. The RDBMS generates a set of unique keys KPT corresponding to the set of probe templates PT, and another set of unique keys KBI for the set of images BIS, which are stored in the NoSQL storage servers determined by using the match score index MSI values.

An essential aspect of enrolment is to perform de-duplication search, which is required to ensure that the user to be enrolled, is not already enrolled in the system. In order to do that, a typical de-duplication search in BAS involves matching the input user data with all of the previously enrolled user's data in the storage. In HDSF, this process is changed by matching the input user data, with only a small subset of data from the total enrolled dataset, which provides performance improvement over the traditional de-duplication search operation in BAS. In order to perform the above task, the following three processes for de-duplication search are proposed:

- Biographic Match Score based Key Filtering,
- Multi-modal Biometric Index based Key Filtering, and
- Key based Biometric Matching.

Every incoming de-duplication search request is handled by the above three processes as shown in Figure 4.7. A de-duplication search request contains the following inputs: biographic data BD, a set of input probe templates PT, biographic decision threshold BDT, indexing threshold IT and decision threshold DT values, whose roles are explained

while discussing the three processes. The purpose of the first two processes is to filter a set of user data from the overall dataset, which is further used in the third process for matching. The details of the three processes are discussed as follows:

Biographic Match Score based Key Filtering: This approach is specifically proposed for enrolment process in HDSF, where each individual field of the input biographic data BD such as first name, last name, department, company and postal code of a user is matched with those of the biographic dataset of the records BDR stored in RDBMS, using Levenshtein edit distance approach [59]. The individual biographic match score BMS obtained by matching each biographic fields in BD and BDR as explained in Appendix 1, are added together to generate a biographic fused score BFS.

$$\text{BFS} = \sum_{i=1}^{n} \text{BMS}_i, \text{ where n = number of biographic fields in BD and BDR}$$

The BFS is further compared with a biographic decision threshold BDT value obtained by the client application with the enrolment request. The keys KBD_R belonging to records BDR, whose biographic fused score BFS after matching with BD is less than or equal to BDT, are returned by the RDBMS to the Storage Configuration module to be further used in *Key based Biometric Matching* process. These keys KBD_R collectively form a set of keys KBD.

$$\text{KBD}_\text{R} \in \text{KBD such that BFS}(\text{KBD}_\text{R}) < \text{BDT}$$

The *Biographic Match Score based Key Filtering* process results in including the keys of the user identities whose biographic fused score BFS values are below the biographic decision threshold BDT value. There is a possibility that the set of keys KBD may or may not contain the keys for the matching records. However, in both the cases, the inclusion

of the keys KBD will not affect the accuracy of the system, as the actual biometric matching performed during the *Key based Biometric Matching* process will filter out the non-matching records and will consider only the one which matches with the input probe templates PT sent during de-duplication search request.

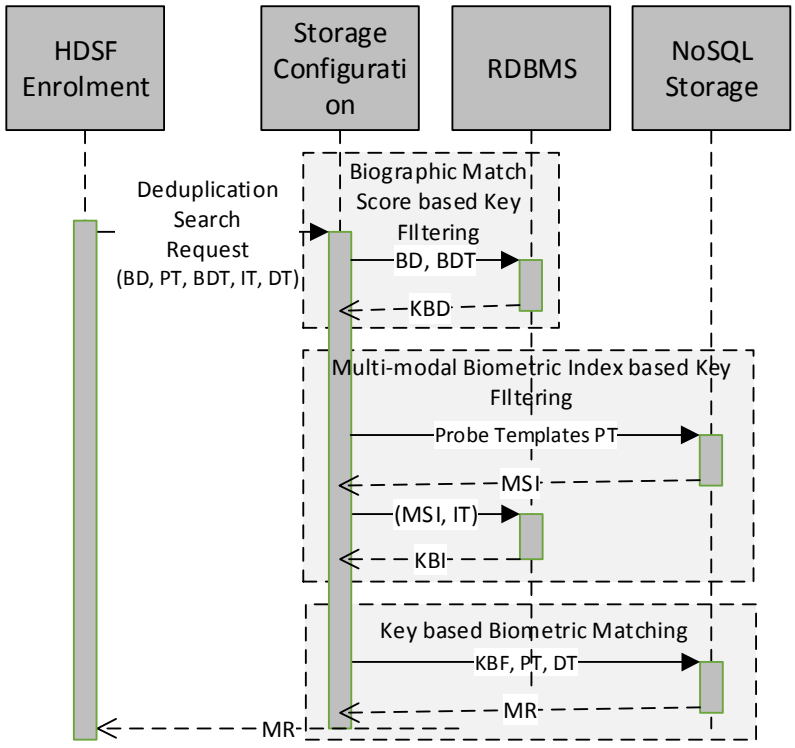

Figure 4.7: Proposed Key Filtering and Biometric Matching Processes

Multi-modal Biometric Index based Key Filtering: Apart from filtering the keys using *Biographic Match Score based Key Filtering* process, another set of keys are filtered based on the match-score index values generated for every biometric template obtained from the data of the enrollee:

1. For each input probe template PT_i^s (s = sub-modality, i = template count in the input template set PT) of a user, an MSI_i^s is calculated in the same way as explained in step 4 of Index Profile Creation and Data Storage process.

2. The value of MSI_i^s is used to obtain the biometric indexing keys KBI_R^s (s denotes the sub-modality, R corresponds to the record of a user such that $R \in$ EI (Total number of enrolled individuals in HDSF)) from RDBMS which fall in the range $\left(MSI_i^s - IT\ ,\ MSI_i^s + IT\right)$, where IT is the indexing threshold value for the sub-modality s. The indexing threshold IT value is any positive real number which is used to filter a set of templates from all templates in the storage, and is provided with the enrolment request by the client application. The IT is further used during performance evaluation of HDSF in section 5.3 to determine the performance improvement at different IT values.
3. The above two steps are repeated for all the probe templates $PT_i^s \in PT$ in order to obtain a set of keys KBI^s for different sub-modalities.
4. These set of keys KBI^s corresponding to different sub-modalities, are used to obtain a final set of keys KBI, which will be used during the Key based Biometric Matching process. There could be two different approaches to obtain the final set of keys:
    a. An intersection of the keys belonging to each set KBI^s is performed to obtain the final set of keys KBI to be used for matching. This provides an additional efficiency improvement by further filtering those keys which do not belong to different sets of KBI^s.

$$KBI = \bigcap_{s=1}^{S} KBI^s \quad \text{where S = total number of sub-modalities}$$

    b. A union of the keys belonging to each set KBI^s is performed to obtain the final set of keys KBI to be used for matching. This may provide a lesser

efficiency improvement than the previous intersection approach but may result in providing lower False Rejection Rate due to the additional keys, improving the overall performance of HDSF.

$$KBI = \bigcup_{s=1}^{S} KBI^{s} \text{ where S = total number of sub-modalities}$$

5. The final set of keys KBI, obtained either from intersection or the union process, is used further in *Key based Biometric Matching* process. The overall performance improvement results obtained by both intersection and union operations are presented separately in section 5.3.

Key based Biometric Matching: All the records in the RDBMS which have similar biographic data to the input biographic dataset such that BFS < BDT, will have a higher probability to be a match. Therefore, during a de-duplication search operation, the keys corresponding to biographic dataset matching KBD (obtained by *Biographic Match Score based Key Filtering* process) are used together with the keys KBI (obtained by *Multi-modal Biometric Index based Key Filtering* process), to form a final set of keys KBF where $KBF = KBD \cup KBI$. Only the set of templates RT corresponding to KBF are matched during the matching process, in place of matching all the templates in order to provide performance during a de-duplication search. The detailed process flow is explained as follows:

1. The set of keys KBF along with probe templates PT and decision threshold DT value are sent to the NoSQL storage to perform matching of PT with the records associated with the keys KBF as shown in Figure 4.7.

2. The match results MR_J for individual match operations are compared with the decision threshold DT value, and are returned as a set of match results MR such that $MR_J \in MR$. The parallel match operation performed by multiple servers inside NoSQL storage also contributes to the overall performance improvement in HDSF during enrolment, as well as identification and verification processes discussed in the following sections.

The combined algorithm for the above three processes is given as follows:

% Multi−modal Biometric Index based Key Filtering and
% Biographic Match Score based Key Filtering processes
for each $PT_i^s \in PT$ % Create MSI for the set of templates
MSI_i^s = Match each PT_i^s with RF_i^s %Obtain match-score index value
% Create a set of match score indexes
$MSI = \bigcup_{i=1}^{n} MSI_i^s$ where n = total number of match score index values
end
$KBF = KBD \cup KBI$ such that MSI_i^s-IT<KBI<MSI_i^s+IT % obtain the keys from RDBMS
% Key based Biometric Matching process
MR_J = Perform Matching and Comparison(PT, RT(corresponding to KBF), DT)
$MR = \sum_{J=1}^{N} MR_J$ % Match results from different servers are combined to form
% the final set of match results MR
where s represents sub-modality and
i corresponds to the particular image or a match score index value count

As shown in the algorithm, a set of match score indexes MSI is created by matching each probe template PT_i^s (s = sub-modality, i = template count in the set BTS) with the corresponding Reference Template $RF_i^s \in RF$. The set of match score indexes MSI is used to obtain a set of keys KBI from RDBMS during *Multi-modal Biometric Index*

based Key Filtering process. Another set of keys KBD obtained by *Biographic Match Score based Key Filtering* process is combined with KBI to generate the final set of keys KBF. During *Key based Biometric Matching* process, the set of keys KBF is used to identify the set of records RT which are matched with the set of input probe templates PT to obtain the match results MR_J from different servers. The match results MR_J from different servers are combined together to form the final set of match results MR.

By using the above discussed proposed processes, a subset of all records enrolled in a biometric system is only selected for actual biometric matching, which contributes to the matching efficiency and eventually to the overall performance improvement during de-duplication search operations. The actual matching efficiency improvement at different indexing threshold IT values is presented in section 5.3.1.

The enrolment in HDSF uses the above four proposed processes: *Index Profile Creation and Data Storage*, *Biographic Match Score based Key Filtering*, *Multi-modal Biometric Index based Key Filtering*, and *Key based Biometric Matching* processes. Therefore, the subsystems and their roles in a traditional BAS are replaced and shared respectively, among different subsystems in HDSF. As shown in Figure 4.8, the role of a BAS controller during enrolment is shared among the API, HDSF Enrolment and Storage Configuration modules, the BAS Storage is replaced by RDBMS and NoSQL Storage, the BAS Template Extractor is replaced by the HDSF Template Extractor module, and the BAS Template Matcher and Match Decision module functionality is handled by the NoSQL storage. The detailed process flow involving the functionalities of different sub-systems during enrolment is explained as follows:

1. An enrolment request sent by a client application to the API layer is routed to HDSF Enrolment module, which contains the biographic data BD of a user, a set of his/her biometric images BIS, a biographic decision threshold BDT value, the decision threshold DT value and the indexing threshold IT value. The indexing threshold value is used to retrieve the biometric keys corresponding to records having matching scores falling in the range $\left(MSI_i^s - IT^s , MSI_i^s + IT^s\right)$ where s represents sub-modality and i denotes the count of different match score indexes.
2. The set of biometric images BIS is sent to the HDSF Template Extractor module which returns a set of probe templates PT where each template $PT_i^s \in PT$ corresponds to a biometric sub-modality s.
3. As a next step, a de-duplication search request is sent to the Storage Configuration module inside the Storage and Processing layer. The request contains the set of probe templates PT, BD, BDT, DT and IT along with the set of biometric images BIS.
4. The Storage Configuration module sends the probe templates PT to NoSQL storage in order to obtain a set of match score index MSI. Each server in the NoSQL storage contains the set of reference templates RF (stored during Index Profile Creation) and matches each $PT_i^s \in PT$ with the corresponding reference template $RF_i^s \in RF$, where s represents sub-modality and i denotes the count of different probe or reference templates.
5. The Storage Configuration module then sends the biographic data BD and BDT to the RDBMS in order to obtain the keys KBD of the records which have a biographic matching score with the BD, less than the threshold BDT, as explained

in *Biographic Match Score based Key Filtering* process. Moreover, the MSI along with IT is also sent to the RDBMS in order to obtain the set of keys KBI corresponding to the range of scores (MSI – IT, MSI + IT), as explained in *Multi-modal Biometric Index based Key Filtering* process.

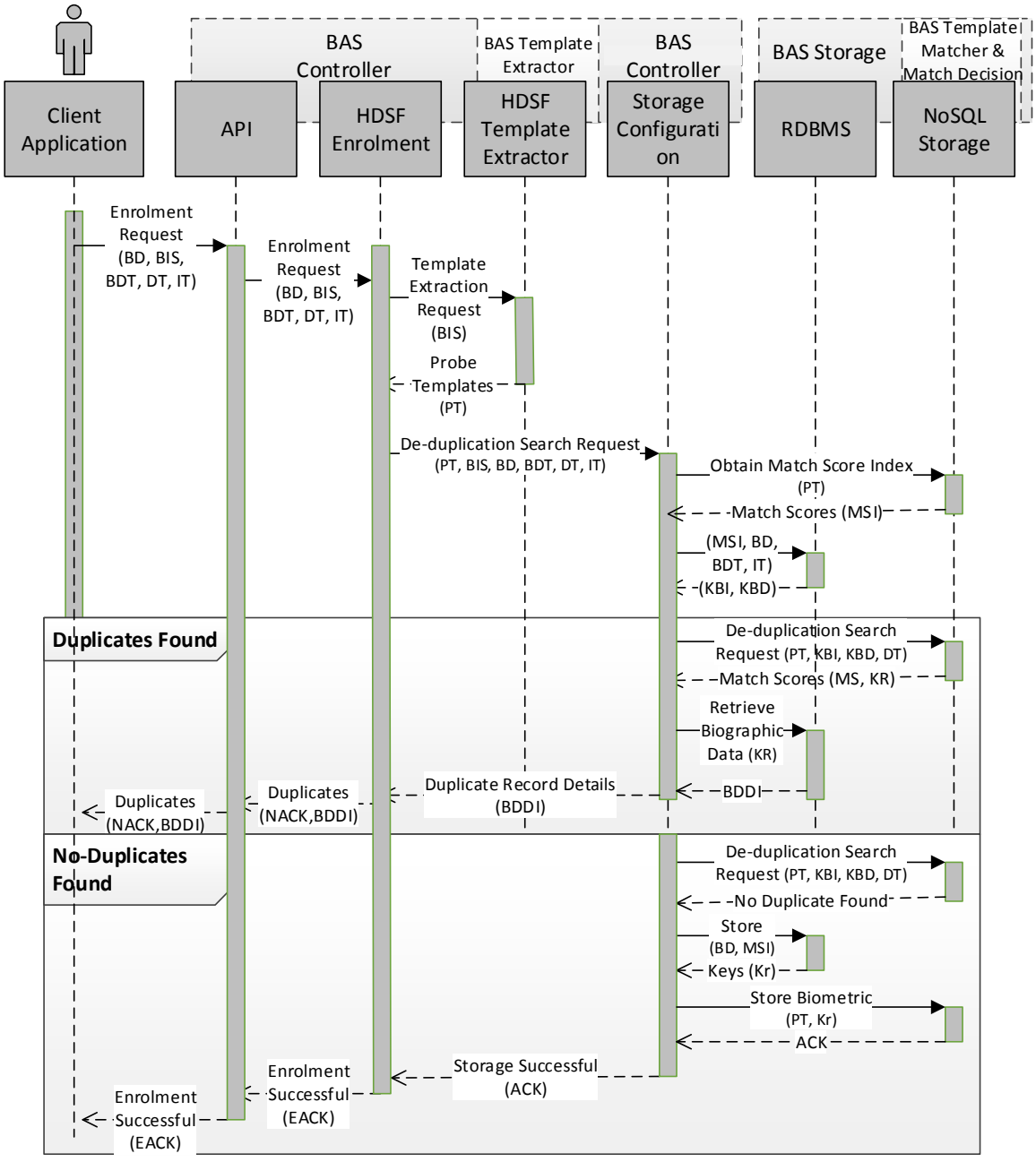

Figure 4.8: Enrolment in HDSF

6. Finally, the set of probe templates PT is sent along with the DT value and the set of keys KBD and KBI, from the Storage Configuration module to the NoSQL storage in order to perform de-duplication search.
7. In case a duplicate record is found during above operation, the NoSQL storage sends back the set of keys KR corresponding to the matching record along with its match scores MS. The keys KR are used to pull the biographic data of the duplicate identity BDDI from RDBMS, which is further sent back to the Enrolment module. The Enrolment module then sends back BDDI along with a negative acknowledgement NACK to the client application through the API.
8. In the case when no duplicate records are found by the NoSQL, it sends back a no duplicate found message to the Storage Configuration module. The Storage Configuration module then stores the biographic details BD obtained as an input with enrolment request, along with MSI values in RDBMS. The RDBMS sends back a set of unique keys K_r corresponding to each template set PT, both of which are stored in the NoSQL storage by using MSI values corresponding to PT. The NoSQL storage sends back a positive acknowledgement ACK response to the Storage Configuration module which sends it back to the Enrolment module. The Enrolment Module then sends an Enrolment success Acknowledgement EACK to the client application through the API.

As discussed, the performance of the de-duplication operation and the overall enrolment process of biometric systems could be improved by using the proposed processes, as only a subset of user data belonging to keys ($KBD \cup KBI$) are matched in contrast to a traditional de-duplication search in BAS involving the matching of all the user data

enrolled in the system. Additionally, all the match operations are performed in parallel inside the NoSQL storage, providing much higher performance than the traditional BAS. The algorithm for enrolment process is given as follows:

% Enrolment in HDSF
PT = Create a set of templates from input images BIS
for each $PT_i^s \in PT$ % Create MSI for the set of templates
MSI_i^s = Match each PT_i^s with RF_i^s %Obtain match-score index value
% Create a set of match score indexes
$MSI = \bigcup_{i=1}^{n} MSI_i^s$ where n = total number of match score index values
end
{KBD,KBI}= Obtain keys from RDBMS using the proposed
filtering processes (MSI,BD,BDT,IT)
KR/No-Duplicate Found = Perform Matching using *Key based*
Biometric Matching process (PT,KBD ∪ KBI,DT)
if(No-Duplicate Found)
K_r = Store in RDBMS(BD,MSI) % Store the BD and MSI in RDBMS
ACK=Store in NoSQL(PT,K_r,MSI) % Store PT with Keys in NoSQL storage
Return ACK
else
BDDI = Retrieve biographic details of duplicate corresponding to KR from RDBMS
Return BDDI

where s represents sub-modality such that $s \in S$(Set of all sub-modalities),
i corresponds to the particular image or template count

As shown in the algorithm, a set of probe templates is created corresponding to the set of input biometric images BIS. Further, a set of match score indexes MSI is created by matching each probe template PT_i^s (s = sub-modality, i = template count in the set BTS) with the corresponding Reference Template $RF_i^s \in RF$. The set of match score indexes

MSI and indexing threshold IT values are used to obtain a set of keys KBI corresponding to *Multi-modal Biometric Index based Key Filtering* process. Another set of keys KBD corresponding to *Biographic Match Score based Key Filtering* process is obtained by using biographic dataset BD and biographic decision threshold BDT value. The set of keys KBI and KBD together with probe templates PT and decision threshold DT value, are used in the *Key based Biometric Matching* process, which could provide either a no-duplicate found result or a set of keys KR corresponding to the duplicates found during the matching process. In the case when no duplicates are found, the biographic data BD along with the MSI values are stored in the RDBMS. The RDBMS returns a set of keys K_r which are stored along with the probe templates PT in the NoSQL storage, using the MSI values. In case a duplicate is found, the biographic data for the duplicate identity BDDI corresponding to the key KR is obtained from the RDBMS and returned to the enrolment module.

4.2.3 HDSF Identification

In HDSF Identification, performance improvement is achieved by using the following two proposed processes:

- Multi-modal Biometric Index based Key Filtering, and
- Key based Biometric Matching.

The details of the above two processes are explained in section 4.2.2, which provide performance improvement during identification search operations in HDSF over those in BAS. By using the above proposed processes, a subset of all records enrolled in a biometric system is only selected for actual biometric matching, which contributes to the matching efficiency and eventually to the overall performance improvement during

identification search operations. The actual matching efficiency improvement at different IT^s values is presented in section 5.3.2. During the identification process in HDSF, the subsystems and their roles in a traditional BAS, are replaced and shared respectively, among different subsystems in HDSF. As shown in Figure 4.9, the role of a BAS controller during identification is shared among the API, HDSF Identification and Storage Configuration modules, the BAS Storage is replaced by RDBMS and NoSQL Storage, the BAS Template Extractor is replaced by the HDSF Template Extractor module, and the BAS Template Matcher and Match Decision module functionality is handled by the NoSQL storage. The detailed process flow involving the functionalities of different sub-systems during identification is described as follows:

1. An identification search request sent by a client application to the API layer is routed to the HDSF Identification module. An identification request typically contains the Biometric Images (BIS) of a user, the decision threshold DT value and the indexing threshold IT value.
2. The set of biometric images BIS is sent to the HDSF Template Extractor module which returns a set of probe templates PT where each template $PT_i^s \in PT$ corresponds to a biometric sub-modality s and template count i in set PT.
3. As a next step, an identification search request containing the set of probe templates PT, BIS, DT and IT, is sent to the Storage Configuration module.
4. The Storage Configuration module sends the probe templates PT to NoSQL storage in order to obtain a set of match score index MSI. Each server in the NoSQL storage contains the set of reference templates RF (stored during Index Profile Creation) and matches each $PT_i^s \in PT$ with the corresponding reference

template $RF_i^s \in RF$, where s represents sub-modality and i corresponds to different probe or reference templates.

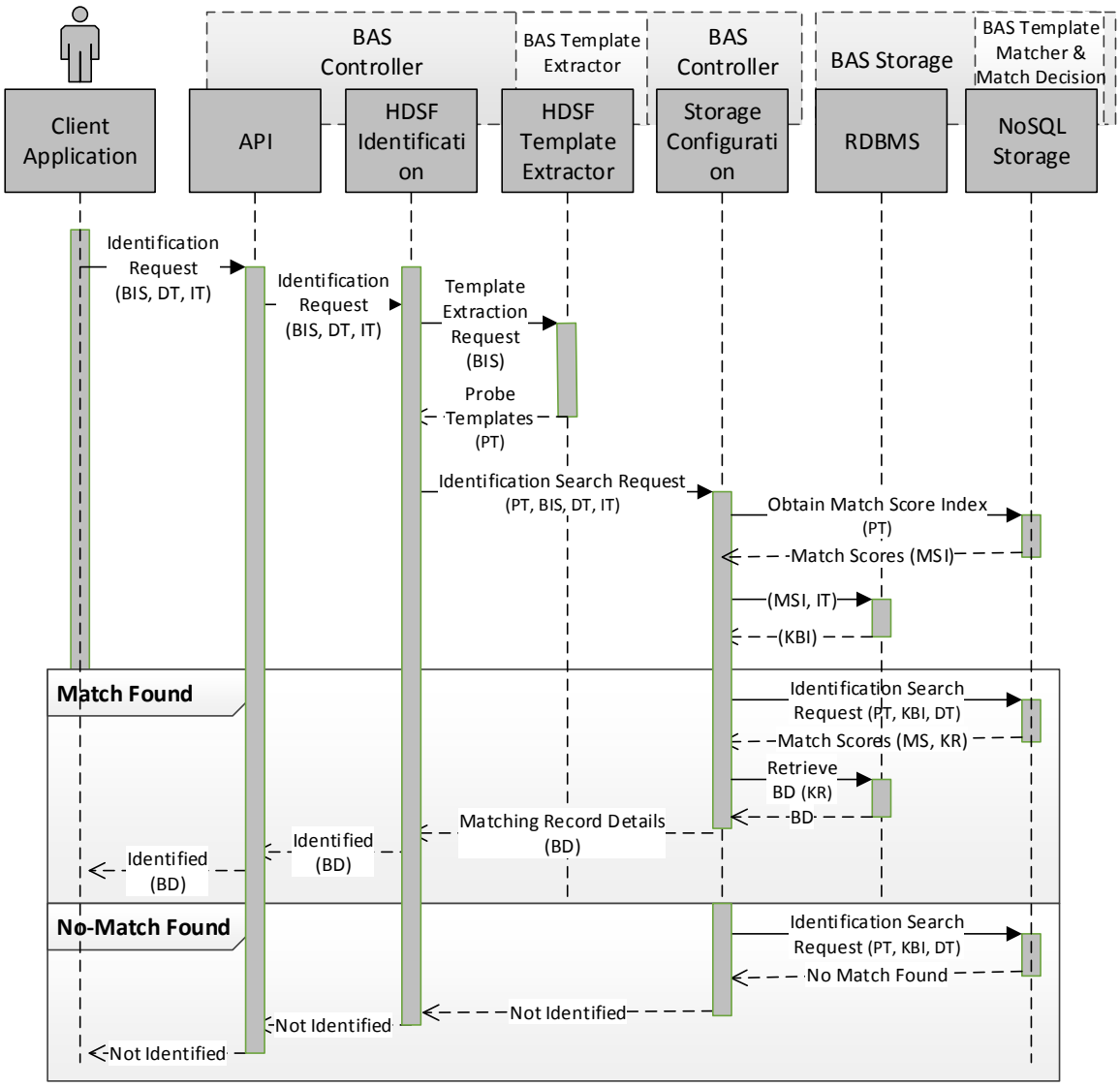

Figure 4.9: Identification in HDSF

5. The Storage Configuration module then sends the MSI along with IT to RDBMS in order to obtain the set of keys KBI corresponding to the range of scores (MSI – IT, MSI + IT).

6. Finally, the set of probe templates PT is sent along with the DT value and the set of keys KBI, from the Storage Configuration module to NoSQL storage in order to perform identification search. The biometric data corresponding to the keys KBI are matched by the match engine (ME) with the PT, in order to search for the matching records.
7. In case a matching record is found during above operation, the NoSQL storage sends back the set of keys KR corresponding to the matching record along with its set of biometric match scores MS. The keys KR are used to pull the biographic details BD of the matching identity from RDBMS, which are further sent back to the Identification module. The Identification module then sends back these details along with an 'Identified' message to the client application through the API.
8. In the case when no matching records are found by the NoSQL, it sends back a no match found message as a null to the Storage Configuration module. The Storage Configuration module sends backs 'Not-Identified' message to the Identification module, which further sends it to the client application through the API. The client application interacting with the HDSF could control the access to a resource based on the two conditions: 'Identified' and 'Not-Identified'.

The algorithm for the identification process is given as follows:

% Identification in HDSF
PT = Create a set of templates from input images BIS
for each $PT_i^s \in PT$ % Create MSI for the set of templates
MSI_i^s = Match each PT_i^s with RF_i^s %Obtain match-score index value
% Create a set of match score indexes
$MSI = \bigcup_{i=1}^{n} MSI_i^s$ where n = total number of match score index values
end

Obtain KBI such that MSI_i^s-IT<KBI<MSI_i^s+IT % obtain the keys from RDBMS
KR/No-Match found = Perform Matching using *Key based Biometric Matching* process (PT,KBI,DT) and obtain keys
if(No-Match found)
    return null
else
    BD = Retrieve biographic data BD of matching records
        corresponding to keys KR from RDBMS
    return BD
where s represents sub-modality such that $s \in$ S(Set of all sub-modalities),
    i corresponds to the particular image or template count, and

As shown in the algorithm, a set of probe templates is created corresponding to the set of input biometric images BIS. Further, a set of match score indexes MSI is created by matching each probe template PT_i^s (s = sub-modality, i = template count in the set BTS) with the corresponding Reference Template $RF_i^s \in RF$. The set of match score indexes MSI and indexing threshold IT values are used to obtain a set of keys KBI corresponding to *Multi-modal Biometric Index based Key Filtering* process. The set of keys KBI together with probe templates PT and decision threshold DT value are used in the *Key based Biometric Matching* process, which could provide either a no-match found result or a set of keys KR corresponding to the matching records found during the matching process. In the case when no matching records are found, a null is returned as the result. In case one or more matching records are found, the biographic data for the matching records BD corresponding to the keys KR are obtained from the RDBMS and returned to the identification module.

4.2.4 HDSF Verification

The purpose of verification is to authenticate a user's identity based on who he/she claims to be. In a typical verification process, a 1:1 comparison is performed between two biometric data: an input probe and a single record stored in the storage, in order to determine whether they belong to the same user. However, in HDSF the verification process is improved over traditional approach in BAS, since more than one verification request could be handled in parallel by the Verification module and other involved sub-systems during the verification process in HDSF.

As shown in Figure 4.10, the role of a BAS controller during verification is shared among the API, HDSF Verification and Storage Configuration modules, the BAS Storage is replaced by RDBMS and NoSQL Storage, the BAS Template Extractor is replaced by the HDSF Template Extractor module, and the BAS Template Matcher and Match Decision module functionality is handled by the NoSQL storage. The detailed process flow involving the functionalities of different sub-systems during verification is described as follows:

1. A verification request sent by a client application to the API layer is routed to the HDSF Verification module. A verification request typically contains the claimed identity CI details of one or more users, a set of Biometric Images (BIS), and a decision threshold DT value. CI could be a set of biographic information associated with one or more users.
2. The set of biometric images BIS is sent to the HDSF Template Extractor module which returns a set of probe templates PT where each template $PT_i^s \in PT$ corresponds to a biometric sub-modality s and template count i in set PT.

3. As a next step, a verification request containing the set of probe templates PT, CI and DT, is sent to the Storage Configuration module.
4. The Storage Configuration module obtains the set of keys K and their associated MSI values from the RDBMS corresponding to the Y number of claimed identities CI_Y such that $CI_Y \in CI$.
5. The set of keys K along with the probe templates PT and decision threshold value DT are send to the NoSQL storage to perform match operations with the records associated with the keys K.

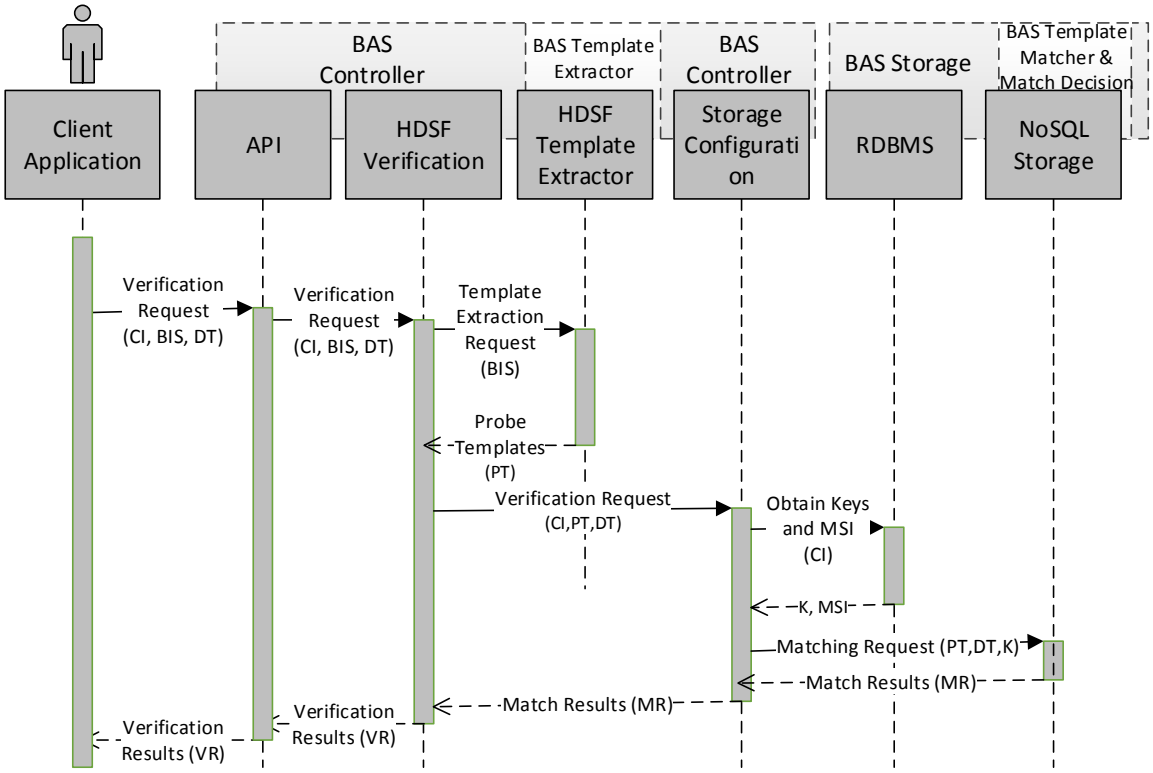

Figure 4.10: Verification in HDSF

6. The set of multiple match results MR, for Y number of claimed identities, are obtained from different servers. The set of match results MR contains individual match/no-match decisions for each claimed identity CI_Y in the set CI. These

match results MR are sent back to the HDSF Verification module which sends it back as the set of verification results VR, to the client application though API.

The client application interacting with HDSF could use the match/non-match decision for various purposes such as access control to a facility or a system, or for the purpose of forensic analysis. The algorithm for the verification process is explained as follows:

%Verification in HDSF
PT = Create a set of templates from input images BIS
{K,MSI}=Select keys corresponding to Y number of claimed Identities ($CI_Y \in CI$)
MR =Perform Matching using *Key based Biometric Matching*
process (PT, RT(corresponding to K), DT)
Return MR
where s represents sub-modality such that $s \in$ S(Set of all sub-modalities), and
i corresponds to the particular match score index value count

As shown in the algorithm, a set of probe templates is created corresponding to the set of input biometric images BIS. Further, the claimed identity details CI associated with one or more claimed identities Y are used to retrieve the set of keys K and a set of match score index MSI values corresponding to the identities Y. The set of keys K together with probe templates PT and decision threshold DT value are used in the *Key based Biometric Matching* process, which provides a set of match results MR. The set of match results MR is returned as a result of the verification process to the verification module.

4.3 Storage and Processing Layer

A hybrid, horizontally scalable, data storage approach is proposed in this research, which is used by the Storage and Processing layer inside HDSF as shown in Figure 4.11. It could efficiently store the different biographic and biometric datasets, and serve BBM layer requests for data access and processing. The storage and processing layer imposes

no restriction over the number and type of underlying data stores and databases. For instance, an application requiring a document storage and graph database would have both the types of storage. In the existing architecture employing biometric data storage, a relational database is used to store biographic data since it is required to have the functionalities such as indexing and querying on the biographic data. Also, most of the existing end-to-end biometric solutions are based on the relational storage due to the same reason. However, the existing biometric systems possess bottleneck in terms of scalability while dealing with biometric datasets. Therefore, the architecture uses a NoSQL type of storage for storing biometric data including images and templates, which provides a scalable storage. Also, the relational database used to store biographic datasets, stores the set of keys for linking the biographic and biometric data of different users.

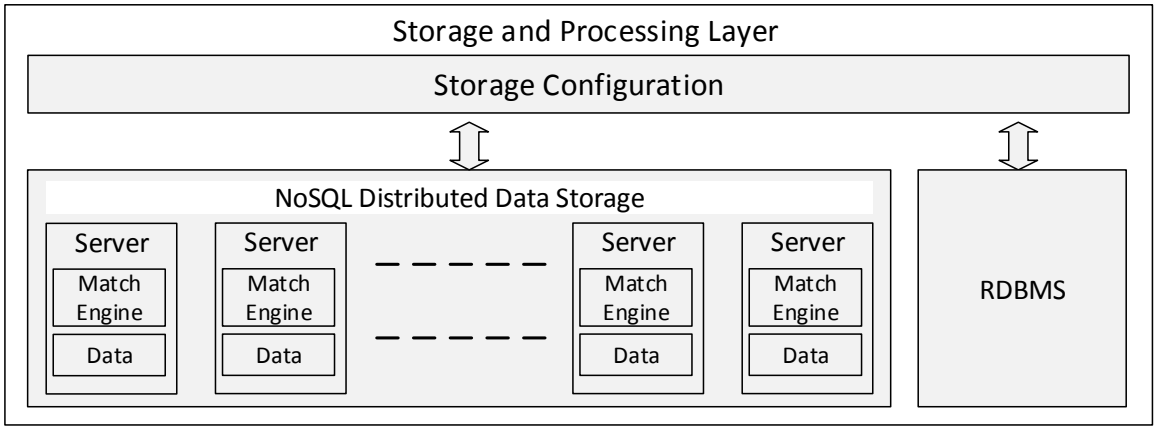

Figure 4.11: Storage and Processing Layer

The Storage and Processing layer provides a seamless integration of the relational model and NoSQL data stores, attaining the benefits of both. It comprises of the following sub-systems: Storage Configuration module, NoSQL Distributed Data Storage and RDBMS.

Overall, the Storage and Processing layer provides the following benefits over the storage provided by a traditional BAS:

- Each server inside the NoSQL Distributed Data Storage provides local data processing on each individual node. The NoSQL storage is provided with an underlying concept of moving computation close to the data rather than moving the data between servers for processing. This is especially beneficial when the size of data set is large and moving it requires large network bandwidth. However, on the other hand, moving computation close to the dataset minimizes network congestion and increases the overall throughput of the system. Therefore, the network bandwidth requirements inside Storage and Processing layer are significantly less as compared to those in traditional data processing systems, where the required data is read from the storage before processing. This also improves the overall performance of the system due to less number of data transfer operations between servers.
- The NoSQL Distributed Data Storage performs parallel matching of biometrics data through separate Match Engines inside different computation nodes, providing a significant improvement in overall performance.
- The Storage and Processing layer is designed to store very large biometric datasets reliably due to the underlying NoSQL storage which also provides horizontal scalability for storage as the dataset size increases.
- The Storage and Processing layer could efficiently store and manage biographic datasets by using RDBMS.

The functionality for each of the Storage and Processing layer sub-system is explained as follows:

4.3.1 Storage Configuration

The storage configuration module handles different enrolment, identification and verification requests as described in sections 4.2.2 - 4.2.4. In order to serve those requests, it performs the following operations:

- Manages the data flow between different RDBMS and NoSQL storage during different biometric processes as shown in Figure 4.7.
- Handles the task of Index Profile Creation as described in section 4.2.2.

In HDSF, the biographic data is stored in the RDBMS whereas the biometric data is stored in the key-value storage.

4.3.2 Relational DBMS

In HDSF, the relational database holds the biographic dataset pertaining to a relational model, providing an easy transition from the existing biometric applications [17]. The biographic dataset includes data belonging to the enrolled identities, relationships between different data, match score index for different templates, and the keys required to retrieve biometric data stored in the NoSQL Distributed Data Storage. The keys which are stored in both RDBMS and NoSQL Distributed Data Storage provide a link between the biographic data stored in the RDBMS and the biometric data stored in the NoSQL Distributed Data Storage. On the other hand, the match score index MSI decides the location of a particular biometric data inside the NoSQL Distributed Data Storage. The

role of RDBMS during the course of Enrolment, Identification and Verification was explained in sections 4.2.2 - 4.2.4.

4.3.3 NoSQL Distributed Data Storage (NDDS)

The NoSQL Distributed Data Storage is used for storing biometric datasets which require massive scalability. Moreover, NDDS provides concurrent and fast read/write access, and local processing over biometric data. NDDS is composed of several low-cost commodity servers where each server stores biometric templates along with a unique key generated by RDBMS, associated with each template. The biometric data is distributed evenly across different servers where each server processes its own set of data using Match Engines as shown in Figure 4.11. The NDDS partitions the biometric data based on MSI of different biometric templates as explained in section 4.2.2.

Match Engines (ME)

In addition to storage, each server could perform its own processing over the local data, using Match Engine (ME) module inside each of them. In HDSF, a match engine specifically performs two processes: **(i) biometric matching** and **(ii) decision**. The two processes are described as follows:

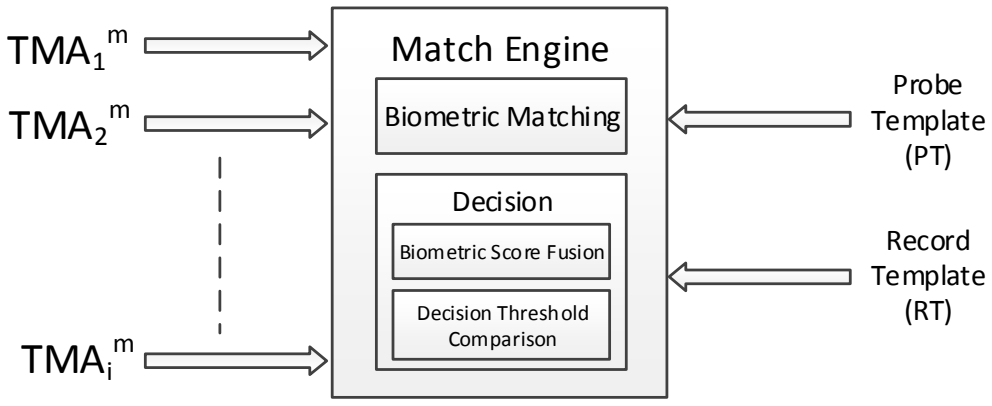

Figure 4.12: Template Matching Algorithm Selection

Biometric Matching: An *adaptive multi-modal biometric matching* approach is proposed in this research which is used by the match engine for on-the-fly algorithm selection of template matching algorithms. The approach determines the suitable template matching algorithm based on the sub-modality s of the probe template PT_i^s in the set of templates PT for each matching operation with the corresponding record template RT_i^s in the set RT. As shown in Figure 4.12, different template matching algorithms forming a set of algorithms $TMA = \{TMA_1^m, TMA_2^m, ..., TMA_i^m\}$, where m denotes the modality for each algorithm and i denotes the algorithm count, are used by the Match Engine to perform matching based on the modality of each probe template in PT. Each biometric match operation between the probe template PT_i^s and the corresponding record template RT_i^s results in a match score MS_n value where n corresponds to the number of match scores generated by matching operations between the probe templates in PT and record templates in RT.

Decision: The decision process of a match engine is further responsible for two operations: biometric score fusion and decision threshold comparison. The details of the two operations are as follows:

1. Biometric Score Fusion: It involves fusing the biometric match scores MS_n corresponding to two different modalities or sub-modalities obtained from the biometric matching process as explained above. As the scores generated for the two different sub-modalities (belonging to the same modality) are often generated by the same template matching algorithm, therefore, they are averaged together to generated an intra-modal fused score FS_m (m represents modality) However, in

case of the scores obtained from different modalities, a more complicated score level multi-modal fusion technique based on Support Vector Machines [60], [61], quality dependent analysis [62], [63], or weighted-sum [64] is needed to be employed in order to obtain sufficient accuracy. The HDSF uses weighted sum technique [64] to calculate the multi-modal fused score; however, any of the above mentioned fusion technique could be used as a biometric fusion function BF() to generate the multi-modal fused score FS. The biometric score fusion process combining the fused score between two iris images and a face image using two different fusion functions BF1() and BF2() is shown in Figure 4.13.

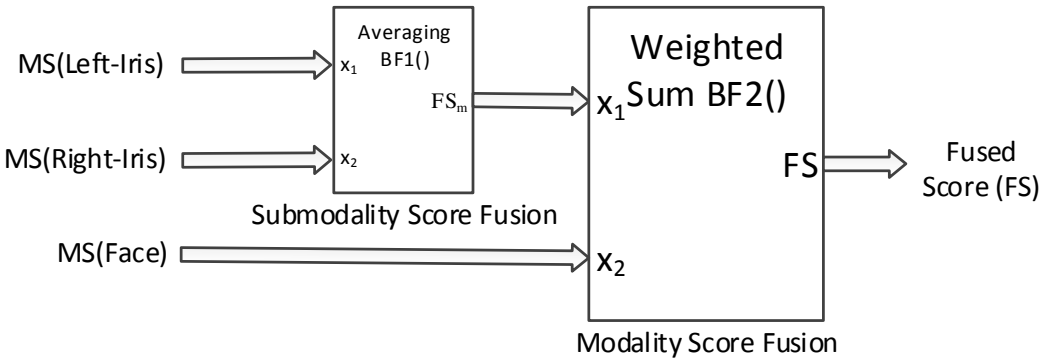

Figure 4.13: Biometric Score Level Fusion in HDSF

2. Decision Threshold Comparison: The decision threshold comparison process uses a decision function to compare the value of the fused score with the decision threshold DT value. It has a different functionality during the verification process than during identification and enrolment processes. In case of verification, the Decision function (D_v) is used which takes the multi-modal fused score FS and DT value as the input and returns back the binary match result MR such as:

$$MR = D_v(FS, DT), \text{ where } MR = \begin{cases} 1, & \text{if } FS \geq DT \\ 0, & \text{if } FS < DT \end{cases}$$

On the other hand, during identification and enrolment, a set of record keys is also sent as an input to the Decision function (D_i) which performs a similar threshold comparison operation as in verification, for the set of fused scores FSS. However, dissimilar to verification, FSS here contains match scores between more than two records due to 1:N comparison between probe and records such that $FS_i \in FSS$ where FS_i is a multi-modal fused score for a particular user. As a final output from the decision module, the subset of keys KR corresponding to all user records having fused scores $FS \geq DT$ value, are returned to the Identification or Enrolment module, while all other keys are discarded.

$$\{MS, KR\} = D_i(FSS, DT)$$

As discussed above, a match engine adopts different approaches during de-duplication and identification search than during verification; therefore, different Match Engine algorithms are used during these processes. The Match Engine algorithm for identification and de-duplication search is shown as follows:

%Match Engine Algorithm for Identification and de-duplication
for each (PT_i^s in PT) %Obtain match scores using biometric matching process
MS_n = Generate Match Score for each match operation between PT_i^s and
RT_i^s (corresponding to keys KI)
end
Obtain fused score FS for each user record using biometric score fusion process
{MS,KR} = Obtain matching scores and keys using Decision Threshold Comparison process
Return KR and MS
where s represents sub-modality such that $s \in S$(Set of all sub-modalities),
i corresponds to the template count in PT or RT, and
n represents the number of match operations generating different scores.

As shown in the algorithm, during de-duplication and identification search a 1:N matching is performed and a match score MS_n is generated for each match operation between the probe templates PT_i^s and the record template RT_i^s corresponding to the input keys KI. The match score MS_n values are used to obtain fused score FS for each user record using the biometric score fusion process. The fused scores FS for each user record are compared with the decision threshold DT value using the decision threshold comparison process, to obtain the set of matching record keys KR along with the corresponding match scores MS.

In contrast to the identification and de-duplication search processes, a verification process involves a 1:1 matching between the templates of a single record and probe, for one or more users. The result comprises of only a set of match results MR containing match/no-match decision for each user record. The Match Engine algorithm for verification is shown as follows:

```
%Match Engine Algorithm for Verification
for each (PT_i^s in PT)    %Obtain match scores using biometric matching process
        MS_n = Generate Match Score for each match operation between PT_i^s and
                RT_i^s (corresponding to keys KI)
end
Obtain fused score FS for each user record using biometric score fusion process
MR =  Obtain matching results using Decision Threshold Comparison process
Return MR
where s represents sub-modality such that s ∈ S(Set of all sub-modalities),
        i corresponds to the particular template in PT or RT, and
        n represents the number of match operations generating different scores.
```

As shown in the algorithm, the biometric matching and biometric score fusion processes are same in the identification/de-duplication and verification algorithms. However, the

difference between the two algorithms is during the decision threshold comparison process where a set of match results MR comprising of match/no-match decision for each user record are returned as an output during verification whereas a set of match scores in keys are returned during identification and de-duplication search opeations.

Overall, NDDS provides the following benefits to the existing architecture:

1. It provides parallel match operations on each server during de-duplication and identification search operations, and could process multiple verification requests simultaneously.
2. It provides horizontally scalable data storage for massive biometric datasets, supporting extensibility for other data types in future.

4.4 Summary

In this chapter, a hybrid, horizontally scalable storage was proposed to store massive biometric datasets, along with storing the associated biographic data. Moreover, a set of four processes were proposed which led to the performance improvement during de-duplication and identification search operations. Also, two additional approaches for adapting different biometric algorithms during run-time were also proposed in this chapter. Further, the underlying architecture of the proposed Hybrid Data Storage Framework was explained. The proposed framework is designed to cater the needs of existing and future biometric systems handling massive datasets with the underlying architecture providing the storage for both structured and unstructured datasets. The processes involved with the biometric algorithms are explained in the beginning, following with a discussion on the API layer serving as an interface to the framework.

Further, the biometrics domain specific functionalities possessed by the Biometric Biographic Management Layer were explained, together with the detailed functionalities of its sub-systems and their interaction with other layers. Finally, the Storage and Processing layer providing a highly scalable storage was discussed in detail; simultaneously, presenting a view towards leveraging the framework by the existing applications based on relational databases.

Chapter 5

5 Implementation & Evaluation

In this chapter, the implementation of Hybrid Data Storage Framework (HDSF) is presented, preceded by a discussion of biometric algorithms and test datasets used in the implementation. The different functionalities exposed by the API layer are shown in the form of API methods exposed as a Windows Communication Foundation (WCF) service. This is followed by a discussion of the implementation of different subsystems inside Storage and Processing layer in which the storage process and matching processes are specifically highlighted. Finally, a section including the evaluation of HDSF is explained highlighting the significant performance improvements over traditional BAS during identification, enrolment and verification processes.

5.1 Biometric Algorithms and Test Datasets

The biometric modalities considered for the evaluation are Face and Iris. The details regarding the biometric algorithms, and test datasets for both biometric and biographic data, are discussed in the following sections:

5.1.1 Face

The face extraction and matching algorithms used for the evaluation are of VeriFace, obtained as a trial license from Neurotechnology. The VeriFace extraction algorithm has an average extraction time of 270 milliseconds generating an average template size of 36 KB; whereas the VeriFace matching algorithm has an average matching time of 4

milliseconds. The test dataset for faces was generated by combining face images obtained from multiple sources such as CASIA [65], Youtube [66] and other [67] open source databases available on the internet.

5.1.2 Iris

The iris extraction and matching algorithms used for the evaluation are of VeriEye, obtained as a trial license from Neurotechnology. The VeriEye extraction algorithm has an average extraction time of 156 milliseconds with an average template size of 3 KB; whereas the VeriEye matching algorithm has an average matching time of 1 millisecond. Similar to face, the test dataset for iris was generated by combining iris images obtained from multiple sources such as CASIA [65] and MMU [68].

5.1.3 Biographic Dataset

The biographic dataset used for the evaluation was created using the tool made available by Generate Data [69]. The tool was used to generate gender independent data containing the following fields: First Name, Department, Organization and Postal Code.

A final dataset containing data for 1738 user identities was created by combining the different face, iris and biographic datasets. The data for each user identity contains biographic fields along with multiple biometric images and templates for face, left-iris and right iris. Out of the total dataset, one image for each modality and its associated sub-modalities was considered for enrolment in the database, whereas the other images were used as probe data for performance evaluation of the system during identification, enrolment and verification processes.

5.2 HDSF Implementation

The implementation of HDSF is explained under different sections, which mention the details about individual layers in HDSF as given below:

5.2.1 API Layer Implementation

The API layer is implemented as a Windows Communication Foundation (WCF) service which exposes different methods for the functionalities provided by HDSF such as: enrolment, identification and verification requests. Moreover, it provides methods to obtain Template Extraction Algorithm (TEA) specific information and select different algorithms based on different application requirements. The API methods for different processes are given as follows:

Enrolment Request

bool EnrolPerson(string[] BiographicData, byte[][] BiometricImages, int BiographicDecisionThreshold, int BiometricDecisionThreshold, float IndexingThreshold, out string DuplicatePersonBiographicDetails)

Identification Request

bool IdentifyPerson(byte[][] BiometricImages, int BiometricDecisionThreshold, float IndexingThreshold, out string IdentifiedPersonBiographicDetails)

Verification Request

bool VerifyPerson(byte[][] BiometricImages, int BiometricDecisionThreshold, string ClaimedIdentityDetails, out string[] VerificationResults)

Template Extraction Algorithm Details Request

bool GetBiometricAlgorithmDetails(out string BiometricAlgorithmDetails)

Template Extraction Algorithm Selection Request

bool SetBiometricAlgorithm(int AlgorithmID, int Modality ID)

Reference Image Enrolment Request

bool EnrollReferenceImage(string[] BiographicData, byte[][] BiometricImages)

Index Profile Creation and Data Storage Request

bool CreateIndexProfile(string[] BiographicData, byte[][] BiometricImages)

5.2.2 BBM Layer Implementation

The different modules inside the BBM layer are responsible for managing the process flow during different operations such as template extraction, enrolment, identification and verification. The process flows for the above operations have already been explained in section 4.2; therefore, the tool developed during the research to send client requests for performing the enrolment, identification and verification processes is shown in Figure 5.1. A user could send the biographic and biometric data along with other parameters such as biographic decision threshold, biometric indexing and decision threshold values using this tool. The enrolment, identification and verification requests send by this application are handled by the HDSF implementation running on a different machine, which processes these requests and sends back the results to the client application shown in Figure 5.1.

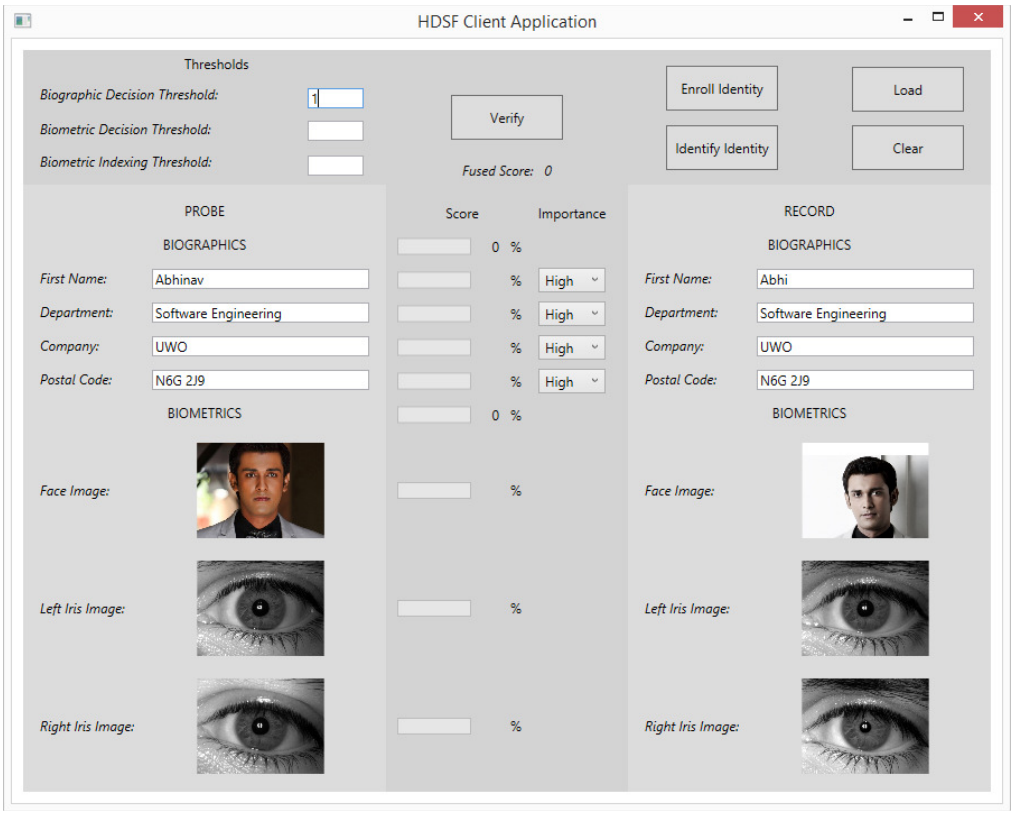

Figure 5.1: HDSF Client Tool

5.2.3 Storage and Processing Layer Implementation

This layer implements the storage and data processing functionalities for HDSF. The details about the implementation for each of its sub-systems are explained as follows:

Relational Database Management System

The Relational DBMS used during the evaluation is MySQL server. As shown in Figure 5.2, MySQL is used to store the biographic information and the details about the biometric dataset such as:

- Data related to different biographic fields
- Associations of different biometric images with the biographic data
- Association between different images and their templates

- Modality and sub-modality details, and
- The keys associated with each image and template.

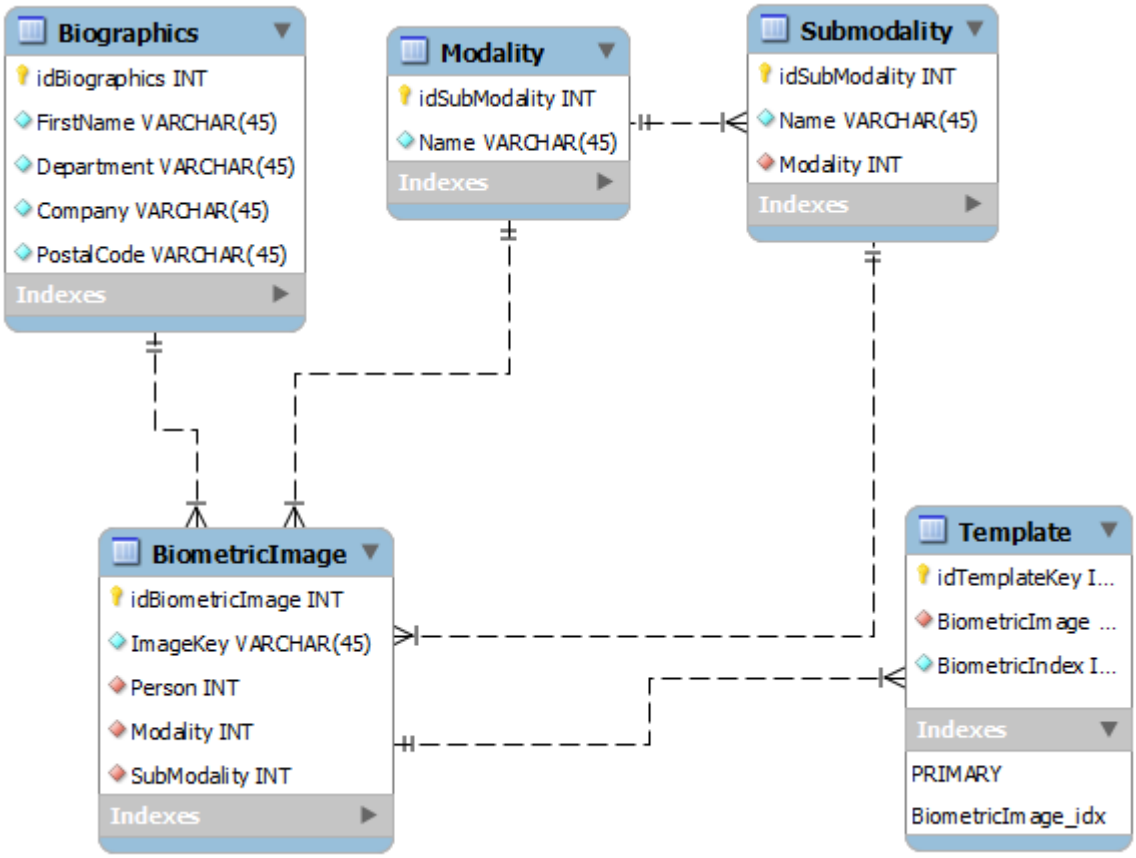

Figure 5.2: RDBMS Schema for HDSF

The biometric images and their associated templates are stored in the NoSQL storage by using the index creation process explained in section 4.2.2.

NoSQL Storage

The NoSQL storage consists of 4 Redis key-value storage instances, each responsible for storing and managing one of the four data partitions. The storage was created using a Windows 7 machine with core-i7 2.0 GHz processor having 4 cores. Moreover, each

Redis instance uses a dedicated match engine to perform parallel matching operations on the data inside its respective data partition.

Storage Configuration

The Storage Configuration module is responsible for serving enrolment, identification and verification requests. In order to serve those requests, the Storage Configuration module performs data access and manipulation by accessing the underlying NoSQL Distributed Data Storage and RDBMS storage as discussed in section 4.2. The Storage Configuration module performs the operations of index creation and data storage, where it stores the biographic data inside MySQL based RDBMS storage and biometric images and templates inside Redis based NoSQL storage. It also performs the matching operation for templates during Enrolment, Identification and Verification operations.

5.3 Evaluation

In this section, the results are obtained by running multiple tests to analyse the impact of the proposed processes on the overall efficiency and accuracy of HDSF. The details of different tests are discussed in the subsequent sections; however, it is important to highlight the different contributing factors which provided performance improvement during different processes. Therefore, the factors responsible for improving the performance are as follows:

- Index Profile Creation and Data Storage:
- Biographic Match Score based Key Filtering,
- Multi-modal Biometric Index based Key Filtering, and
- Key based Biometric Matching.

An important point to note is that the purpose of the second and third processes out of the above four processes, is to filter a subset of keys in order to reduce the total number of individual match operations. This filtering process could lead to false rejection of genuine records in case a genuinely matching record is not present in the subset of keys obtained after filtering. This leads to an increase in the False Rejection Rate (FRR) of the overall system. However, there will be no impact on the False Acceptance Rate (FAR) since the filtering of keys could only reject a genuine person, but could not contribute to the false acceptance of impostors, as the actual matching of biometric data will detect those impostors. Therefore, while comparing performance improvement during different processes, the respective FRR value is also calculated for each indexing threshold IT value in order to optimize the overall system for best matching efficiency versus FRR trade-off. Also, as the number of filtered records depends upon the indexing threshold IT value and the match score index MSI value for the particular probe, the results are obtained by varying the IT values. The indexing threshold IT values are increased from a low initial value till the FRR reaches to value of zero. This is done in order to analyse the overall matching efficiency improvement at different accuracy levels governed by FRR. Since, it was not possible to manually send each input probe data while running these tests, an index based matching tool was developed during the research as shown in Figure 5.3 to automate the above process of obtaining the performance results (matching efficiency versus FRR) for different indexing threshold values.

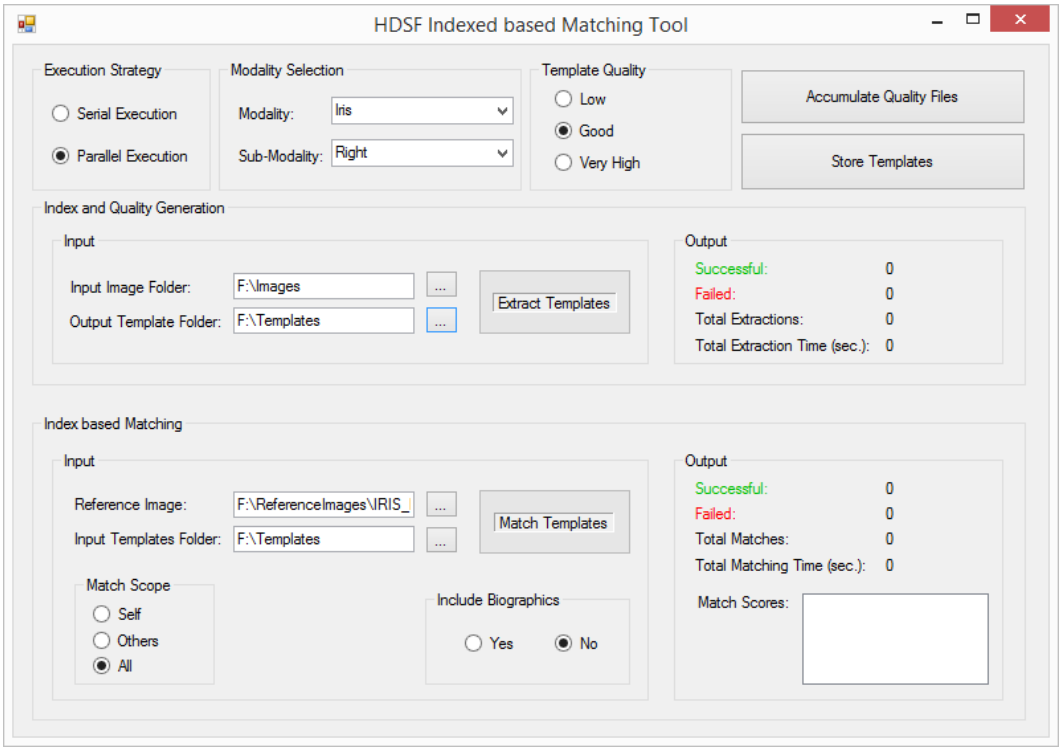

Figure 5.3: HDSF Index Based Matching Tool

5.3.1 Matching Efficiency Improvement during HDSF Enrolment

During the process of enrolment in HDSF, there is a contribution of the following proposed processes:

- Index Profile Creation and Data Storage:
- Biographic Match Score based Key Filtering,
- Multi-modal Biometric Index based Key Filtering, and
- Key based Biometric Matching.

The set of keys used during the matching process are KBD (obtained by *biographic match score based key filtering*) and KBI (obtained by *multi-modal biometric index based key filtering*). However, as discussed in section 4.2.2, the set of keys KBI could be obtained either by performing intersection of the keys from different sub-modalities or taking the union of all the keys belonging to different sub-modalities. Therefore, different

results were obtained by applying the two different intersection and union based approaches. The results pertaining to intersection and union based approaches are shown in Table 5.1 and Table 5.2, respectively. Furthermore, the set of keys KBD obtained due to the biographic matching performed between the biographic data of the probe (BD) and those of the records stored in MySQL server, could be different for different biographic data threshold BDT values. However, running the tests for multiple BDT values was not important as a high BDT value would result in a decreased performance with lesser contribution towards the Genuine Acceptance Rate (GAR) of the overall system. A value of 4 was considered keeping into account a maximum of 4 character error in the overall biographic data of the person. Therefore, the final set of keys used during a de-duplication search in an enrolment process was:

$$\text{Total Keys} = \text{KBD} \cup \text{KBI},$$

where, KBI could be due to intersection or union based approach

The small value of BDT ensures that it does not contribute to a decline in the matching efficiency by including a large number of extra keys for matching. However, at the same time it helps in improving the FRR by including those keys which may be incorrectly filtered out by *multi-modal biometric index based key filtering* process. The following tables show the matching efficiency improvement during an enrolment process over a traditional BAS, when adopting an intersection based approach for biometric keys:

Table 5.1: Matching Efficiency Improvement versus FRR during HDSF Enrolment (Intersection of Biometric Keys)

Indexing Threshold (*IT*)	**Matching Efficiency Improvement in HDSF (over BAS) %**	**False Rejection Rate (FRR) %**
0.005	88.4	82.9
0.01	82.9	69.9

0.015	77.4	58
0.02	72.1	48.6
0.025	66.8	39.7
0.03	61.8	32.1
0.035	56.9	25.9
0.04	52.3	20
0.045	47.9	15.5
0.05	43.7	11.1
0.055	39.6	8
0.06	35.7	5.6
0.065	32.1	3.1
0.07	28.7	1.9
0.075	25.6	1.1
0.08	22.9	0.6
0.085	20.2	0.2
0.09	17.8	0.1
0.095	15.6	0

The results in Table 5.1 highlight a matching efficiency improvement of more than 15% at the indexing threshold IT value of 0.095, with zero FRR. However, in a practical system, an FRR value falling in the range of 1 - 3.5% [70], [71] is often considered to be acceptable in terms of overall accuracy of the system. Therefore in the acceptable range of FRR using our dataset, HDSF could provide a matching efficiency improvement of more than 32% over traditional BAS systems for the indexing threshold value of 0.065, resulting in higher overall performance gain during enrolment.

Furthermore, a different set of results as shown in Table 5.2 were obtained by conducting tests which adopted the union based approach for biometric index based key filtering. The results obtained by the union based approach using our dataset provided significant efficiency improvement over the intersection based approach. As shown in Table 5.2, the FRR values dropped more rapidly as compared to the intersection based approach, resulting in a much higher efficiency improvement of more than 51% at zero FRR value,

at an indexing threshold value of 0.04. Moreover, matching efficiency improvement of more than 57% could be obtained in HDSF over a traditional BAS mentioned in section 2.4, using our dataset with the union based approach and in the acceptable range of FRR values, i.e. up to 3.5% as shown in Table 5.2. After comparing the results obtained for both intersection and union based approaches as shown in Figure 5.4, it could be deduced that the union based approach should be adopted over intersection based approach while performing biometric index based key filtering process to obtain the set of keys KBI.

Table 5.2: Matching Efficiency Improvement versus FRR during HDSF Enrolment (Union of Biometric Keys)

Indexing Threshold (*IT*)	**Matching Efficiency Improvement in HDSF (over BAS) %**	**False Rejection Rate (FRR) %**
0.005	87.5	77.9
0.01	81.6	58.0
0.015	76.1	41.0
0.02	71.1	28.6
0.025	66.	17.3
0.03	60.7	8.0
0.035	56.3	1.2
0.04	51.5	0

These performance improvements obtained in HDSF over BAS as shown in Table 5.1 and Table 5.2 should be considered as minimum, since those were obtained by using a Windows 7 machine whose performance was limited by the four cores operating in parallel. In a real application, a much higher efficiency improvement could be obtained by employing a large number of dedicated servers performing parallel match operations, as parallel matching is one of the contributing factors in performance improvement during *Key based Biometric Matching* process.

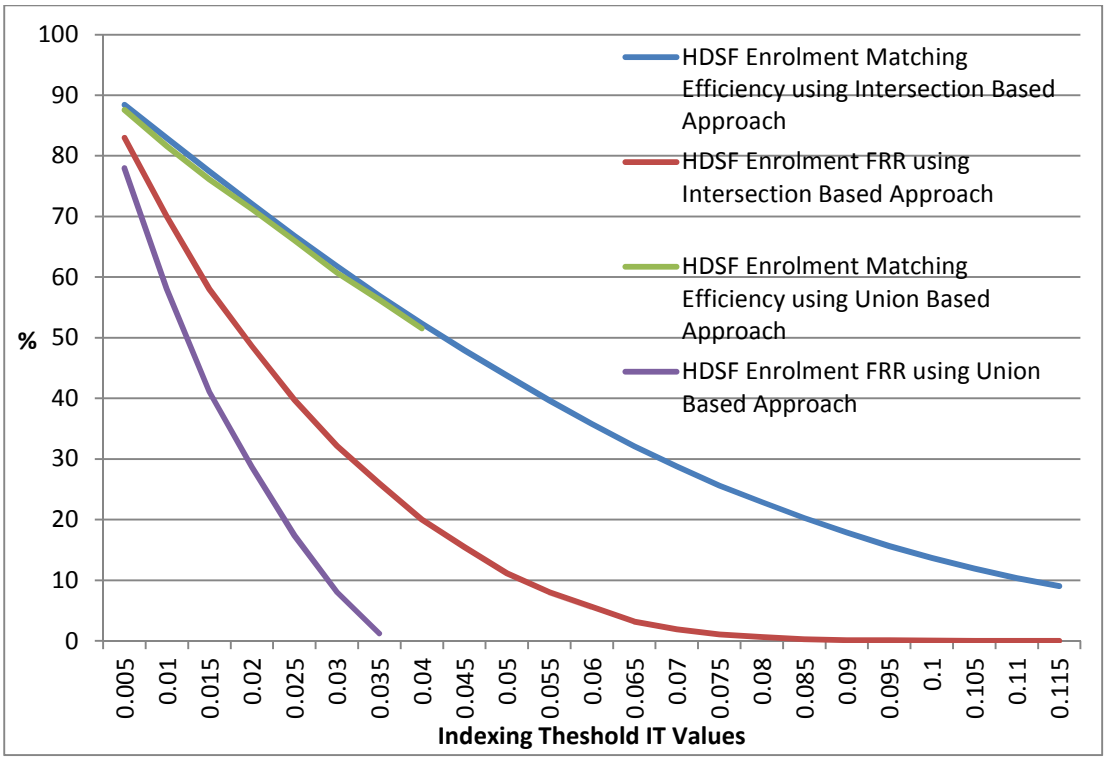

Figure 5.4: Performance Comparison between Intersection and Union Based Approaches during HDSF Enrolment

5.3.2 Matching Efficiency Improvement during HDSF Identification

In an identification search, there is a contribution of the following proposed processes in order to provide performance benefits:

- Multi-modal Biometric Index based Key Filtering, and
- Key based Biometric Matching.

In contrast to enrolment, matching efficiency improvement in HDSF is achieved due to filtering of records only based on *multi-modal biometric index based key filtering* process. Moreover, similar to enrolment process, the set of KBI could be obtained based on the intersection and union based approaches as discussed in section 4.2.2. Therefore, different tests were performed using the two intersection and union based approaches,

whose results are provided in Table 5.3 and Table 5.4, respectively. The following table shows the matching efficiency improvement during an identification process in HDSF over a traditional BAS using our dataset, for different indexing threshold IT values:

Table 5.3: Matching Efficiency Improvement versus FRR during HDSF Identification (Intersection of Biometric Keys)

Indexing Threshold (*IT*)	**Matching Efficiency Improvement in HDSF (over BAS) %**	**False Rejection Rate (FRR) %**
0.005	91.9	84.4
0.01	86.4	71.3
0.015	80.8	59.5
0.02	75.4	50.1
0.025	70	41
0.03	64.9	33.3
0.035	59.9	27.0
0.04	55.2	21.1
0.045	50.6	16.6
0.05	46.2	12.3
0.055	41.9	9.1
0.06	38	6.7
0.065	34.3	4.3
0.07	30.8	3
0.075	27.6	2
0.08	24.7	1.4
0.085	21.9	0.7
0.09	19.4	0.5
0.095	17.2	0.3
0.1	15.1	0.2
0.105	13.4	0.1
0.11	11.8	0

The following table shows the results for the union based approach used to obtain filtered keys KBI:

Table 5.4: Matching Efficiency Improvement versus FRR during HDSF Identification (Union of Biometric Keys)

Indexing Threshold (*IT*)	Matching Efficiency Improvement in HDSF (over BAS) %	False Rejection Rate (FRR) %
0.005	91.6	78.9
0.01	86	59.4
0.015	80.5	42.6
0.02	75.1	30.1
0.025	69.6	18.6
0.03	64.5	9.2
0.035	59.6	2.1
0.04	54.8	0

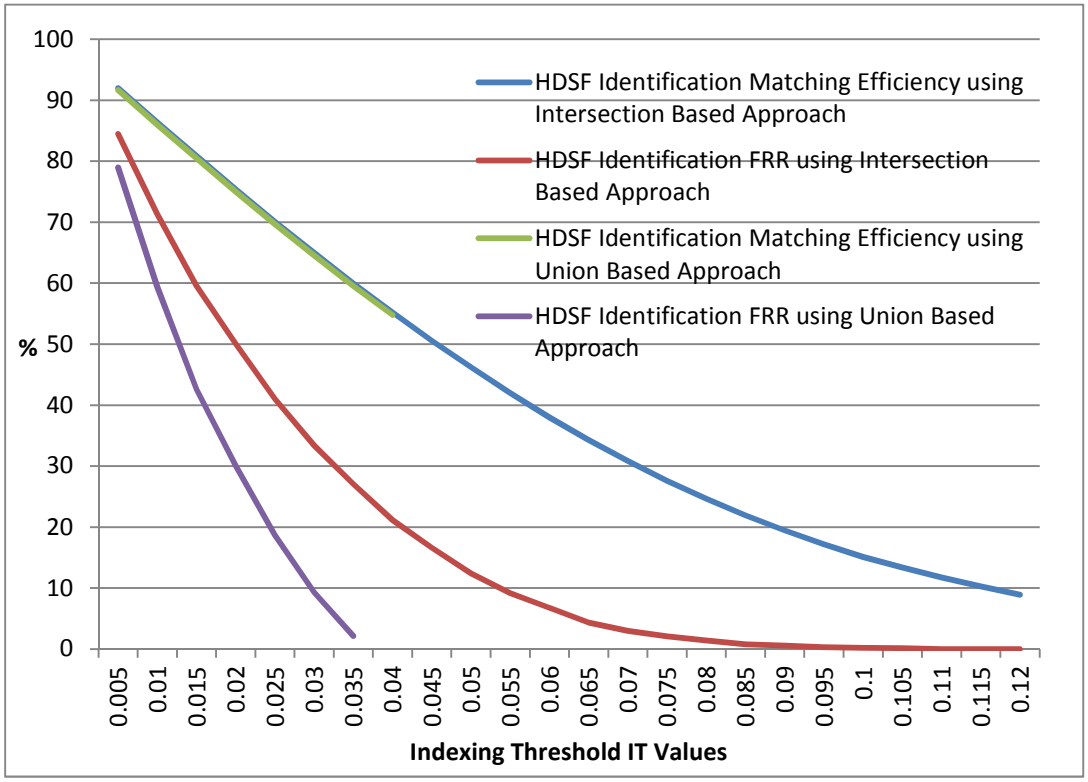

Figure 5.5: Performance Comparison between Intersection and Union Based Approaches during HDSF Identification

As shown in Table 5.4, a matching efficiency improvement of more than 54% for zero FRR at an indexing threshold value of 0.04 could be obtained in HDSF over a traditional BAS mentioned in section 2.4, using our dataset with the union based approach.

Moreover, the HDSF could provide more than 60% matching efficiency improvement over BAS using our dataset during identification process in the acceptable range of FRR values, i.e. up to 3.5%. Similar to enrolment, the union based approach for biometric index based key filtering provided much higher i.e. 54 - 60% matching efficiency improvement in HDSF, over the intersection based approach which provided a lesser 11 - 31% improvement in matching efficiency over the acceptable range of FRR as shown in Figure 5.5. Moreover, as mentioned earlier, the achieved performance improvements should be considered as minimum, since during the evaluation a four core Windows 7 machine were used. In a real application, a larger number of dedicated matching servers with multiple cores would be able to provide much higher performance improvement than those achieved during the evaluation in this research.

5.3.3 Performance Improvement Comparison between HDSF Identification and HDSF Enrolment

A matching efficiency improvement due to Biographic Match Score based Key Filtering process could be obtained by comparing the results for the HDSF Enrolment and HDSF Identification, for both the intersection and union based approaches. For intersection based approach, a comparison of the results in Table 5.1 and Table 5.3 is shown in Figure 5.6; whereas, for the union based approach the comparison of results in Table 5.2 and Table 5.4 is shown in Figure 5.7. For both of these approaches, a matching efficiency improvement of more than 1% was obtained for all the FRR values by the inclusion of biographic match score based key filtering process in de-duplication search operations during enrolment, over the identification search operations.

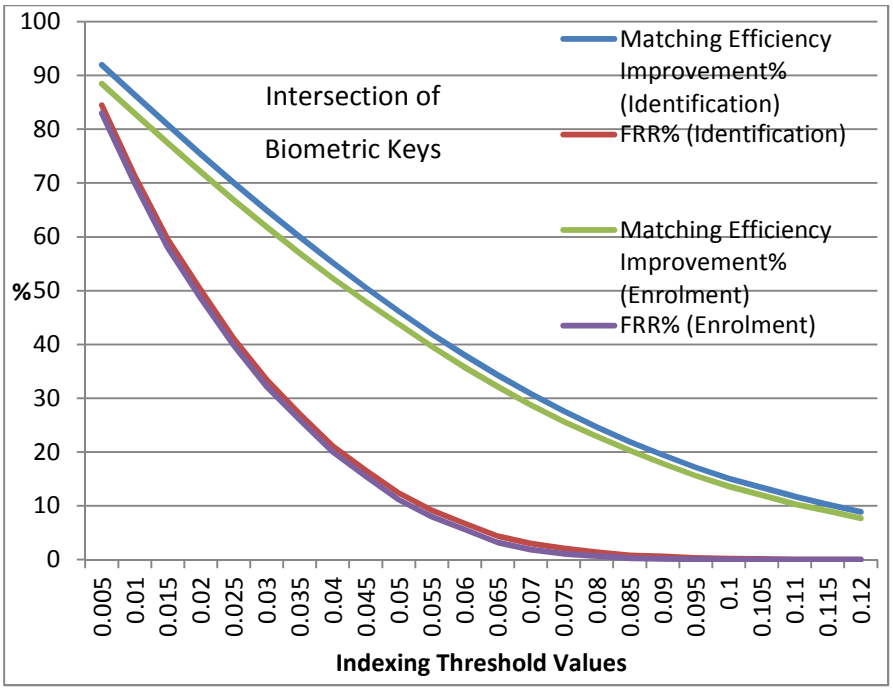

Figure 5.6: Performance Improvement Comparison between HDSF Enrolment and HDSF Identification using Intersection Based Approach

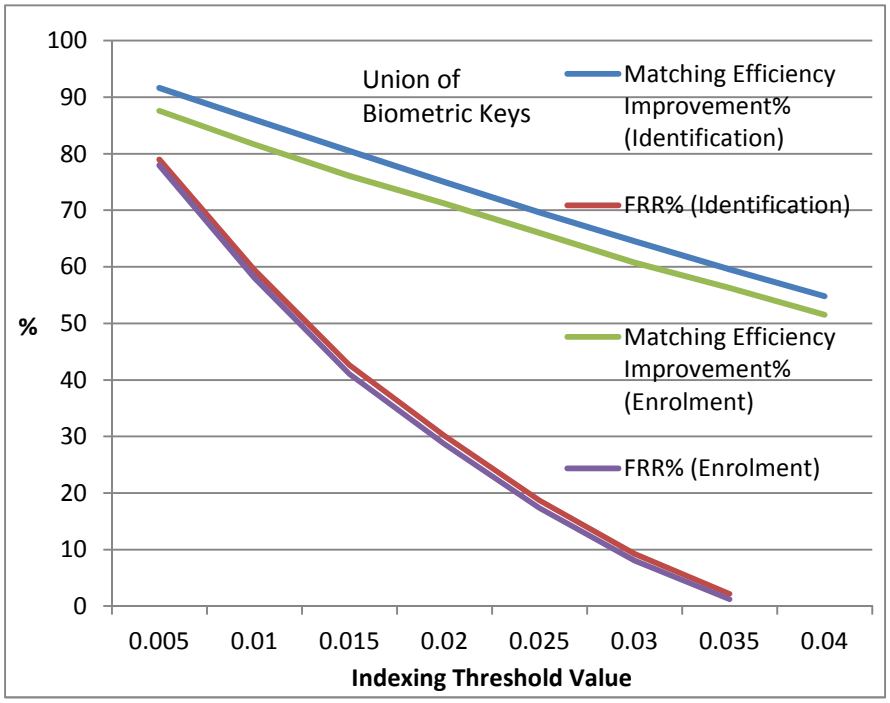

Figure 5.7: Performance Improvement Comparison between HDSF Enrolment and HDSF Identification using Union Based Approach

Moreover, as shown in Figure 5.6 and Figure 5.7., the overall reduction of around 0.05% in FRR was also achieved at different threshold values, which is obvious as some of those records mistakenly considered as false matches by biometric index matching were also considered as genuine match pairs. Therefore, the inclusion of biographic match score based key filtering process resulted in an additional performance gain of 1% in HDSF within the acceptable range of values for FRR.

5.3.4 Performance Improvement during HDSF Verification

One of the contributing factors during performance improvement in HDSF is the parallel match operations performed by multiple match engines simultaneously, than those performed serially in a traditional BAS. The parallel matching contributes to the performance improvement during de-duplication, identification and verification processes in HDSF. During de-duplication and identification processes, other processes such as: *Biographic Match Score based Key Filtering* and *Multi-modal Biometric Index based Key Filtering* processes, also contribute to the overall performance improvement in HDSF. Therefore, evaluating the performance improvement due to parallel matching alone during the de-duplication and identification processes is a complex task. However, parallel matching across multiple match engines is the only contributing factor for performance improvement during verification in HDSF. Therefore, a different test running multiple verification operations was conducted in order to obtain the performance improvement during verification process in HDSF. The test was performed using a single core of a Windows 7 machine termed as serial matching and using four cores of the same Windows 7 machine termed as parallel matching in Figure 5.8, respectively. The results of the test as shown in Figure 5.8 highlights that a parallel

matching provided lower response time for the overall verification test as compared to serial matching operations similar to those performed in a traditional BAS. This lower response time obtained due to parallel matching contributes to the overall performance improvement obtained in HDSF.

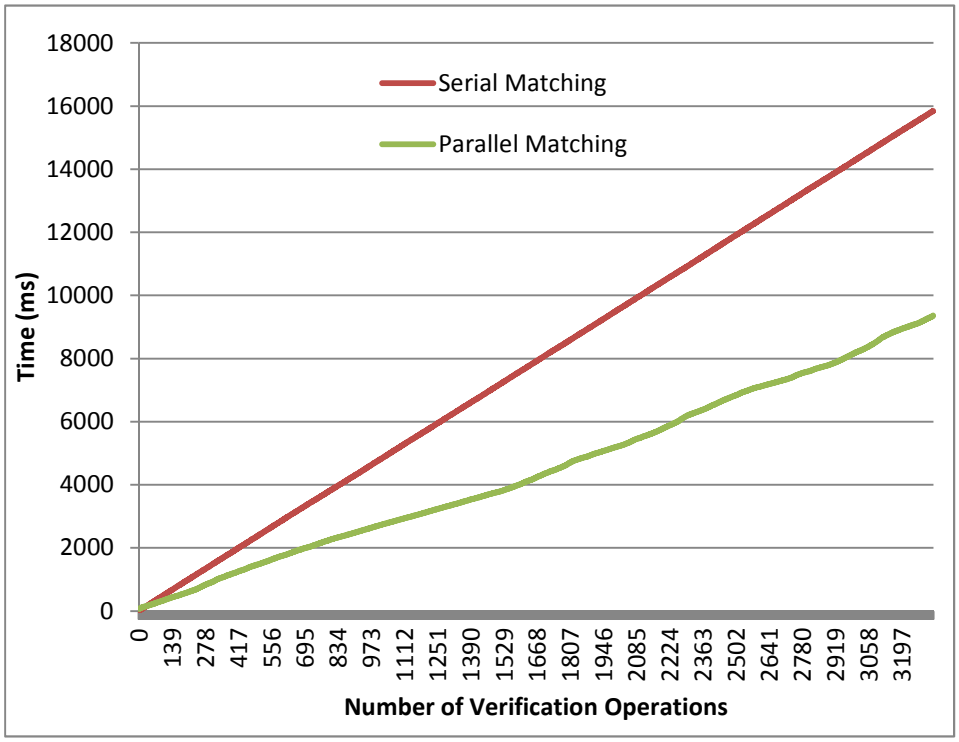

Figure 5.8: Performance Improvement during HDSF Verification

5.4 Rationale behind Performance Improvement in HDSF

The dataset used in the evaluation contains data for 1738 user identities which is larger than those used in most of the previous research studies [17]–[21], [37], but may not be considered as a massive dataset, which typically have data for few hundred thousand or millions of users. Therefore, the following key points need to be considered in order to understand the benefits of HDSF over traditional BAS, while dealing with massive biometric datasets:

- **Overall performance of HDSF improves with larger datasets:** In the existing biometric systems, matching efficiency scales linearly with increasing the number of user records in the storage, as the match operations are performed sequentially. On the contrary, HDSF associates a match-score index value with each biometric template during the index profile creation process. The match-score index value helps in filtering a subset of the entire user records for matching, by using the *Multi-modal Biometric Index based Key Filtering* process proposed in this research and provides a significant improvement in overall performance of HDSF. For instance, the indexing threshold value of 0.035 provides an acceptable FRR as shown in Table 5.2 and Table 5.4. In a case when match-index values for templates are distributed evenly on a scale of 0 to 1, an indexing threshold value of 0.035 will provide a key-filtering range of 0.07 (0.035 X 2). This window of 0.07 on the scale of 0 to 1 will reduce the match space to 7% on an average, as compared to 100% in traditional BAS where all the records are matched. This improvement of 93% in overall match space provides significant reduction in matching time, improving the overall performance of the system. However, as shown in Table 5.2 and Table 5.4, the obtained improvements are less than 93% as there are additional overheads involved while matching such as locating the servers and distributing the keys, retrieving and aggregating results. As the user dataset size increases, these additional overheads does not change much and become less significant as the total matching time is dominated by the 1:N matching operations. Therefore, it could be deduced that as the size of the user

dataset is increased, a higher overall performance improvement could be achieved in HDSF over traditional BAS.

- **Matching could be scaled by increasing parallel match engines:** Another contributing factor for performance improvement in HDSF is the use of parallel match engines associated with separate server cores. It is to be noted that the results provided in Table 5.1 - Table 5.4 are obtained using only 4 cores on a single development machine. However, in a practical application involving large biometric datasets, a much higher performance improvement could be achieved by using a larger number of dedicated servers.

5.5 Summary

In this chapter, it was shown how the proposed processes contributed to the performance improvement in HDSF over traditional BAS. Initially, the implementation of Hybrid Data Storage Framework (HDSF) was presented, which involved implementing the various functionalities of the framework and providing those functionalities to the client applications through the API methods. It was shown how the different layers have been implemented, highlighting the tools and technologies used. Finally, the evaluation of the HDSF was presented specifically highlighting its performance improvement over traditional Biometric Authentication Systems when used with our dataset during the process of enrolment, identification and verification.

Chapter 6

6 Conclusions and Future Work

This chapter presents the conclusion to this thesis through a reflection on the work that has been accomplished. The possibilities of future work are further presented which outline other interesting areas of research that can expand upon this work.

6.1 Conclusions

Biometric Authentication is a desirable approach for access control applications where security of a system is directly dependent upon the accuracy of the authentication mechanism. The accuracy and robustness offered by a BAS provides an edge towards its adoption over the traditional manual or semi-automated approaches for authentication. As a result, a huge number of applications in different domains have started leveraging the benefits offered by a BAS in their system. Moreover, several large-scale identity matching systems have been evolved in recent times which incorporate a BAS for providing services to government and national agencies. However, these large-scale applications have started realizing bottlenecks in terms of scalability due to the large size of their biometric datasets. Furthermore, an increase in the enrolments and the incorporation of multi-modal solutions for increased accuracy, are worsening the scalability related issues. Also, an increase in the size of biometric datasets in these large-scale biometric applications, is adversely affecting their performance in terms of slower recognition rates during identification and de-duplication search operations. This is further resulting in to perform the enrolment as an offline process increasing the risk of

multiple enrolments and affecting the security of the system. Therefore, there is an unavoidable need of a new approach which could provide a scalable storage along with effectively increasing the performance of the overall biometric processes.

The major contribution of this thesis to the biometrics domain is the creation of a Hybrid Data Storage Framework (HDSF). This HDSF is created with the following characteristics:

- The HDSF provides a scalable storage required for large-scale identity matching applications in order to store large biometric datasets. It uses a key-value based NoSQL storage to store biometric images and templates belonging to different users. The storage provides horizontal scalability as opposed to RDBMS which provides vertical scalability or a memory based storage limited towards its size.

- The framework is capable of storing and managing the biographic data associated to different users. It uses a relational database for this purpose which provides indexing and querying over the biographic datasets as opposed to it being a limitations with systems using file system based [18]–[20] and memory based [21] storage.

- HDSF provides performance improvement at multiple levels over a traditional BAS. First, it provides an effective process for *multi-modal biometric index based key filtering* process, for biometric dataset filtering in order to reduce the search space during identification and de-duplication searches. It provides a significant performance improvement as only a subset of biometric records is matched during a search operation as opposed to matching all the records in a BAS. At the next

level, HDSF uses another proposed *biographic match score based key filtering* process, in order to contribute in the performance improvement specifically during the de-duplication search operations in an enrolment process. Finally, at the third level, HDSF scales out the matching process by performing parallel biometric match operations. These match operations are performed by different match engines on separate server instances and provides a significant performance improvement during the different biometric processes of identification, enrolment and verification. It could specifically provide higher performance for the applications requiring multiple verification requests to be served at the same time.

- HDSF provides on-the-fly selection of different biometric algorithms based on different application requirements. This could be very useful for a variety of applications requiring the selection between different algorithms, having different performance in terms of accuracy and efficiency. Providing algorithm selection for each individual biometric process as an option could be a significant asset for wide adoption of the framework by a number of applications.

- HDSF presents a biometric modality independent interoperable framework, providing no limitations towards inclusion of any number and type of biometric modalities. Moreover, it provides the benefits of multi-modal system by effectively using these modalities towards improving the overall performance of the framework.

- HDSF provides an easy migration from a large number of existing biometric systems which are based on RDBMS, by internally providing the relational storage for biographic datasets.

Minor contributions were also made towards the adoption of HDSF by providing the framework as a service through a uniform API layer. This API layer exposes the internal functionalities offered by the HDSF by abstracting the details of the different storages used for biographic and biometric data. Further, this API is exposed as a service in order to enable access through different applications and devices.

The scalability and performance issues associated with large-scale biometric systems could be solved by different ways. For example, the improvement in performance could be achieved by specifically working on each subsystem of a large biometric system. HDSF provides the solution to the foreseeable problems related to scalability and performance by one out of those several different ways.

6.2 Future Work

The issues associated with large-scale biometric systems are numerous and providing the solution for all of them is an ongoing research process. The HDSF provides the solution to the subset of those issues; however the other issues related with biometric systems are yet to be considered in the future research work. Areas that need to be addressed are as follows:

- As biometric data belonging to a user could uniquely identify him/her, therefore, maintaining the security of the biometric data should be one of the most important goals to be achieved by any biometric system [1]. A biometric system should be

able to provide multiple levels of security such as: the security of data at rest, the security of data when moving it in between different internal subsystems and the security of the data while interacting with any external system. In future, HDSF will include additional security mechanisms than those provided by the existing underlying RDBMS and NoSQL storage.

- As with an increase in adoption of biometric systems, the numbers of spoofing attempts to access the biometric systems have also increased simultaneously. Therefore, defining appropriate data and response sharing policies is also an important task, as sharing more than what is required, could be hazardous towards the security of a biometric system. For example, consider a system using multiple biometric modalities matching in order to authenticate a user. The system captures each biometric data and provides a sequential response for each operation to the user. In this case, an impostor attempting unauthorized access through the biometric system could easily determine which biometric modality failed while matching. Later on, he/she could further try to spoof that particular modality again to access the system, while performing the similar spoof attacks for other modalities. Therefore, to avoid this problem, a biometric system like this should not share the individual responses for each modality and must provide the result as a single match/no-match decision by internally combining the match results from different modalities. Therefore, as a future work, the HDSF will try to provide the facility of defining different data and response sharing policies by the applications using HDSF.

- As different biometric modalities are inherently different in terms of accuracy, and so are the different biometric algorithms for different modalities. Moreover, two different algorithms for the same modality could be based on different techniques to perform extraction and matching processes, resulting in different performance and accuracy achieved by them. Therefore, a biometric system leveraging these algorithms should be able to smartly handle these characteristics of different algorithms in order to provide better recognition performances in different scenarios. For example, a less accurate but faster biometric algorithm should be used in a different scenario than a more accurate and slower algorithm. Therefore in future, the HDSF will be improved to be adaptable to different scenarios in run-time based on the different parameters such as probe image quality, user-defined performance requirements and user-defined quality parameters for matching.

In conclusion, HDSF is a significant step towards addressing the scalability and performance issues in large-scale biometric systems. Currently, the existing biometric systems have started showing inefficiencies towards handling of massive biometric datasets, which could be effectively handled by the use of HDSF in those systems. Moreover, HDSF could be easily adopted by a large number of the existing systems, as it could internally provide the relational storage, used by several of them.

Bibliography

[1] S. Ye, Y. Luo, J. Zhao, and S.-C. S. Cheung, "Anonymous Biometric Access Control," *EURASIP Journal on Information Security*, vol. 2009, pp. 1–17, 2009.

[2] R. S. Sandhu and P. Samarati, "Access control: principle and practice," *IEEE Communications Magazine*, vol. 32, no. 9, pp. 40–48, Sep. 1994.

[3] A. K. Jain, P. J. Flynn, and A. A. Ross, "Applications of Biometrics," in *Handbook of biometrics*, Springer, 2008, pp. 12–15.

[4] S. Mahadik, K. Narayanan, D. V. Bhoir, and D. Shah, "Access Control System using fingerprint recognition," in *Proceedings of the International Conference on Advances in Computing, Communication and Control - ICAC3 '09*, 2009, pp. 306–311.

[5] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric recognition: security and privacy concerns," *IEEE Security & Privacy Magazine*, vol. 1, no. 2, pp. 33–42, Mar. 2003.

[6] H.-J. Kim, "Biometrics, is it a viable proposition for identity authentication and access control?," *Computers & Security*, vol. 14, no. 3, pp. 205–214, Jan. 1995.

[7] S. A. Shaikh and J. R. Rabaiotti, "Characteristic trade-offs in designing large-scale biometric-based identity management systems," *Journal of Network and Computer Applications*, vol. 33, no. 3, pp. 342–351, May 2010.

[8] N. W. Jensen and J. D. Gansemer, "US-VISIT Independent Verification and Validation Project: Test Bed Establishment Report No. LLNL-TR-466881.," 2011.

[9] G. Greenleaf, "India's national ID system: Danger grows in a privacy vacuum," *Computer Law & Security Review*, vol. 26, no. 5, pp. 479–491, Sep. 2010.

[10] R. N. Rodrigues, L. L. Ling, and V. Govindaraju, "Robustness of multimodal biometric fusion methods against spoof attacks," *Journal of Visual Languages & Computing*, vol. 20, no. 3, pp. 169–179, Jun. 2009.

[11] Y. Ding and A. Ross, "A comparison of imputation methods for handling missing scores in biometric fusion," *Journal of Pattern Recognition*, vol. 45, no. 3, pp. 919–933, Mar. 2012.

[12] U. D. O. Justice, "The Federal Bureau of Investigation," 2013. [Online]. Available: http://www.fbi.gov/about-us/cjis/fingerprints_biometrics/ngi/ngi2.

[13] F. Alonso-Fernandez, J. Fierrez, and J. Ortega-Garcia, "Quality Measures in Biometric Systems," *IEEE Security & Privacy Biometrics Compendium*, vol. 10, no. 6, pp. 52–62, 2011.

[14] J. J. Romero, "India's big bet on identity," *IEEE Spectrum*, vol. 49, no. 3, pp. 48–56, Mar. 2012.

[15] K. Ricanek Jr. and C. Boehnen, "Facial Analytics: From Big Data to Law Enforcement," *Computer*, vol. 45, no. 9, pp. 95–97, Sep. 2012.

[16] B. DeCann and A. Ross, "'Has this person been encountered before?': Modeling an anonymous identification system," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012, pp. 89–96.

[17] J. R. Díaz-Palacios, V. J. Romo-Aledo, and A. H. Chinaei, "Biometric access control for e-health records in pre-hospital care," in *Proceedings of the Joint Extending Database Technology*, 2013, pp. 169–173.

[18] F. Liu, Q. Zhao, and D. Zhang, “A novel hierarchical fingerprint matching approach,” *Journal of Pattern Recognition*, vol. 44, no. 8, pp. 1604–1613, Aug. 2011.

[19] U. Park and A. K. Jain, “Face Matching and Retrieval Using Soft Biometrics,” *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 406–415, Sep. 2010.

[20] D. Peralta, I. Triguero, R. Sanchez-Reillo, F. Herrera, and J. M. Benitez, “Fast fingerprint identification for large databases,” *Journal of Pattern Recognition*, vol. 47, no. 2, pp. 588–602, Feb. 2014.

[21] G. Danese, M. Giachero, F. Leporati, and N. Nazzicari, “An embedded multi-core biometric identification system,” *Journal of Microprocessors and Microsystems*, vol. 35, no. 5, pp. 510–521, Jul. 2011.

[22] A. K. Jain, P. J. Flynn, and A. A. Ross, “Operations of Biometric Systems,” in *Handbook of biometrics*, Springer, 2008, pp. 3–12.

[23] V. Govindraju, W. Hamilton, J. Hurt, A. A. Ross, C. J. Tilton, and D. M. Waymire, “Biometric System Architecture,” in *IEEE Certified Biometrics Professional - Module 3*, IEEE, 2012, pp. 2–7.

[24] A. K. Jain, A. Ross, and S. Prabhakar, “An Introduction to Biometric Recognition,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, Jan. 2004.

[25] P. Campisi, R. L. Carter, and C. W. Crooks, “Biometric Modalities,” in *IEEE Certified Biometrics Professional - Module 2*, 2012, pp. 10–115.

[26] S. Chauhan, A. S. Arora, and A. Kaul, “A survey of emerging biometric modalities,” *Proceedings of the International Conference on Biometrics Technology*, vol. 2, pp. 213–218, Jan. 2010.

[27] R. M. Bolle, J. H. Connell, and N. K. Ratha, “Biometric perils and patches,” *Journal of Pattern Recognition*, vol. 35, no. 12, pp. 2727–2738, Dec. 2002.

[28] A. K. Jain, K. Nandakumar, and A. Nagar, “Biometric Template Security,” *Journal on Advances in Signal Processing*, vol. 2008, no. 1, pp. 113:1–113:17, 2008.

[29] U. Uludag, A. Ross, and A. Jain, “Biometric template selection and update: a case study in fingerprints,” *Journal of Pattern Recognition*, vol. 37, no. 7, pp. 1533–1542, Jul. 2004.

[30] N. Yager and T. Dunstone, “The Biometric Menagerie,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 2, pp. 220–230, Feb. 2010.

[31] S. Ribari, D. Ribari, and N. Paveši, “Multimodal biometric user-identification system for network-based applications,” *IEE Proceedings - Vision, Image, and Signal Processing*, vol. 150, no. 6, pp. 409–416, 2003.

[32] M. M. Monwar and M. L. Gavrilova, “Multimodal biometric system using rank-level fusion approach.,” *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, vol. 39, no. 4, pp. 867–78, Aug. 2009.

[33] M. Espinoza and C. Champod, “Risk evaluation for spoofing against a sensor supplied with liveness detection.,” *Forensic science international*, vol. 204, no. 1–3, pp. 162–168, Jan. 2011.

[34] S. Modi and E. H. Spafford., “Future Biometric Systems and Privacy,” in *Privacy in America*, Scarecrow Press, 2011, pp. 167–183.
[35] A. Rattani, N. Poh, and F. Roli, “Critical analysis of adaptive biometric systems,” *Journal of Biometrics, IET*, vol. 1, no. 4, pp. 179–187, Dec. 2012.
[36] P. Reid, “An Introduction to Statistical Measures of Biometrics,” in *Biometrics for Network Security*, Prentice Hall PTR, 2004, pp. 204–220.
[37] Q. Tao and R. Veldhuis, “Biometric Authentication System on Mobile Personal Devices,” *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 4, pp. 763–773, Apr. 2010.
[38] S. Z. Li and A. K. Jain, “Soft Biometrics,” in *Encyclopedia of Biometrics*, 2nd ed., Springer, 2009, pp. 1235–1248.
[39] M. Shapiro and E. Miller, “Managing databases with binary large objects,” in *16th IEEE Symposium on Mass Storage Systems in cooperation with the 7th NASA Goddard Conference on Mass Storage Systems and Technologies (Cat. No.99CB37098)*, 1999, pp. 185–193.
[40] C. Rathgeb and A. Uhl, “Iris-Biometric Hash Generation for Biometric Database Indexing,” in *Proceedings of 20th International Conference on Pattern Recognition*, 2010, pp. 2848–2851.
[41] H. Mehrotra, B. G. Srinivas, B. Majhi, and P. Gupta, “Indexing Iris Biometric Database Using Energy Histogram of DCT Subbands,” in *Proceedings of Second International Conference, IC3 2009, Noida, India*, 2009, pp. 194–204.
[42] H. Proenca, “Iris Biometrics: Indexing and Retrieving Heavily Degraded Data,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 1975–1985, Dec. 2013.
[43] U. Jayaraman, S. Prakash, and P. Gupta, “Indexing Multimodal Biometric Databases Using Kd-Tree with Feature Level Fusion,” in *Proceedings of 4th International Conference, ICISS 2008, Hyderabad, India.*, 2008, pp. 221–234.
[44] C. Nance, T. Losser, R. Iype, and G. Harmon, “NoSQL vs RDBMS - Why there is room for both,” in *Proceedings of the Southern Association for Information Systems Conference, Savannah, GA, USA March 8th–9th*, 2013, pp. 111–116.
[45] C. Coronel, S. Morris, and P. Rob, “Transaction Management and Concurrency Control,” in *Database Systems: Design, Implementation, and Management*, Cengage Learning, 2012, pp. 390–412.
[46] S. Philippi, “Model driven generation and testing of object-relational mappings,” *Journal of Systems and Software*, vol. 77, no. 2, pp. 193–207, Aug. 2005.
[47] I. Konstantinou, E. Angelou, C. Boumpouka, D. Tsoumakos, and N. Koziris, “On the Elasticity of NoSQL Databases over Cloud Management Platforms,” pp. 2385–2388, 2011.
[48] R. Hecht and S. Jablonski, “NoSQL evaluation: A use case oriented survey,” in *2011 International Conference on Cloud and Service Computing*, 2011, pp. 336–341.
[49] R. Cattell, “Scalable SQL and NoSQL Data Stores,” *ACM SIGMOD Record*, vol. 39, no. 4, pp. 12–27, 2011.
[50] P. Atzeni, F. Bugiotti, and L. Rossi, “Uniform access to NoSQL systems,” *Information Systems*, pp. 1–17, Jun. 2013.

[51] K. Grolinger, W. A. Higashino, A. Tiwari, and M. A. Capretz, “Data management in cloud environments: NoSQL and NewSQL data stores,” Journal of Cloud Computing: Advances, Systems and Applications.2013, 2:22, DOI: 10.1186/2192-113X-2-22.

[52] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, “Bigtable,” *ACM Transactions on Computer Systems*, vol. 26, no. 2, pp. 1–26, Jun. 2008.

[53] D. Crockford, “Json (javascript object notation),” 2006.

[54] R. M. Lerner, “At the forge: memcached,” *Linux Journal*, vol. 2008, no. 176, 2008.

[55] G. DeCandia, D. Hastorun, and M. Jampani, “Dynamo: amazon’s highly available key-value store,” *SOSP*, vol. 41, no. 6, pp. 205–220, 2007.

[56] B. Calder, H. Simitci, J. Haridas, C. Uddaraju, H. Khatri, A. Edwards, V. Bedekar, S. Mainali, R. Abbasi, A. Agarwal, M. F. ul Haq, J. Wang, M. I. ul Haq, D. Bhardwaj, S. Dayanand, A. Adusumilli, M. McNett, S. Sankaran, K. Manivannan, L. Rigas, A. Ogus, N. Nilakantan, A. Skjolsvold, S. McKelvie, Y. Xu, S. Srivastav, and J. Wu, “Windows Azure Storage,” in *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles - SOSP ’11*, 2011, pp. 143–157.

[57] P. Sanfilippo, Salvatore and Noordhuis, “Redis.” [Online]. Available: http://redis.io. [Accessed: 02-Jan-2014].

[58] R. Klophaus, “Riak Core,” in *ACM SIGPLAN Commercial Users of Functional Programming on - CUFP ’10*, 2010.

[59] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions and reversals,” *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[60] B. J. Kang and K. R. Park, “A new multi-unit iris authentication based on quality assessment and score level fusion for mobile phones,” *Machine Vision and Applications*, vol. 21, no. 4, pp. 541–553, Feb. 2009.

[61] H. F. Liau and D. Isa, “Feature selection for support vector machine-based face-iris multimodal biometric system,” *Expert Systems with Applications*, vol. 38, no. 9, pp. 11105–11111, Sep. 2011.

[62] N. Poh, T. Bourlai, and J. Kittler, “A multimodal biometric test bed for quality-dependent, cost-sensitive and client-specific score-level fusion algorithms,” *Journal of Pattern Recognition*, vol. 43, no. 3, pp. 1094–1105, Mar. 2010.

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

[64] Z. Akhtar, G. Fumera, G. L. Marcialis, and F. Roli, “Evaluation of multimodal biometric score fusion rules under spoof attacks,” in *2012 5th IAPR International Conference on Biometrics (ICB)*, 2012, pp. 402–407.

[65] Casia, “Biometrics Ideal Test,” *National Laboratory of Pattern Recognition, Institute of Automation (NLPR)*, 2010. [Online]. Available: http://www.idealtest.org/findTotalDbByMode.do?mode=Iris.

[66] L. Wolf, T. Hassner, and I. Maoz, “Face recognition in unconstrained videos with matched background similarity,” in *CVPR 2011*, 2011, pp. 529–534.

[67] M. Grgic and K. Delac, “Face Recognition Homepage,” *VCL*, 2013. [Online]. Available: http://www.face-rec.org/databases/.
[68] M. University, “Iris Recognition Homepage,” 2006. [Online]. Available: http://www1.mmu.edu.my/~ccteo/.
[69] B. Keen, “Generate Data,” 2014. [Online]. Available: http://www.generatedata.com. [Accessed: 02-Jan-2014].
[70] C.-L. Lin, T. C. Chuang, and K.-C. Fan, “Palmprint verification using hierarchical decomposition,” *Journal of Pattern Recognition*, vol. 38, no. 12, pp. 2639–2652, Dec. 2005.
[71] M. van der Veen, T. Kevenaar, G.-J. Schrijen, T. H. Akkermans, and F. Zuo, “Face biometrics with renewable templates,” in *Proc. SPIE 6072, Security, Steganography, and Watermarking of Multimedia Contents VIII, 60720J*, 2006, pp. 60720J1 – 60720J12.

Appendix A: Biographic Match Score Calculation

A Biographic matching process consists of matching the different fields of two different biographic datasets BD_1 and BD_2. In order to understand the biographic matching process, let us consider two biographic fields: first name and last name, in each sets of biographic data BD_1 and BD_2. Let us consider the following values for each of the two fields:

BD_1: First Name = Michael BD_2: First Name = Mike

BD_1: Last Name = Doug BD_2: Last Name = Douglas

According to Levenshtein[59], the distance between two strings could be calculated by counting the minimum number of insertions, deletions and substitutions in order to transform one string to another. Mathematically, the distance between two strings a and b could be represented as $\mathrm{lev}_{a,b}(|a|,|b|)$ where,

$$\mathrm{lev}_{a,b}(\mathrm{i},\mathrm{j})=\begin{cases}\max(i,j) & \text{if } \min(i,j)=0\\ \min\begin{cases}\mathrm{lev}_{a,b}(i-1,j)+1\\ \mathrm{lev}_{a,b}(i,j-1)+1\\ \mathrm{lev}_{a,b}(i-1,j-1)+1_{(a_i\neq b_j)}\end{cases} & \text{otherwise}\end{cases}$$

where, $1_{(a_i\neq b_j)}$ is the indicator function equal to 0 when $a_i=b_j$ and 1 otherwise.

Definition A.1: A Biographic Match Score (BMS) is generated by calculating the levenshtein distance between biographic fields of same type belonging to two different user data BD1 and BD2 such that:

BMS_1 = lev(BD_1: First Name, BD_2: First Name) = lev(Michael, Mike) = 4

BMS_2 = lev(BD_1: Last Name, BD_2: Last Name) = lev(Doug, Douglas) = 3

Definition A.2: A Biographic Fused Score (BFS) is obtained by adding the biographic match score (BMS) values obtained by matching the biographic fields of two different user data BD1 and BD2 such that: $\mathrm{BFS} = \sum_{i=1}^{n} \mathrm{BMS}_i$, where n = number of biographic fields in BD_1 and BD_2. Therefore, when applied to the above case, the value of BFS = 7.

Curriculum Vitae

Name: Abhinav Tiwari

Post-secondary Education and Degrees:

- MESc in Software Engineering
  Western University, Canada
  2012-2014

- B.Tech in Electronics and Communication
  Uttar Pradesh Technical University, India
  2001-2005

Related Work Experience:

- Teaching and Research Assistant
  Western University, Canada
  2012-2014

- Team Lead,
  Accenture Research Lab,
  Accenture Global Services, India.
  2010-2012

- Software Design Engineer,
  Embedded Research Lab,
  Industrial Engineering, India
  2005-2010

Publications:

- K. Grolinger, W. A. Higashino, **A. Tiwari**, and M. A. Capretz, "Data management in cloud environments: NoSQL and NewSQL data stores," Journal of Cloud Computing: Advances, Systems and Applications.2013, 2:22, DOI: 10.1186/2192-113X-2-22.

- **Abhinav Tiwari**, Aleem Haji, Alexandra L'Heureux, Wesley Hunt, Miriam Capretz, Rupinder Mann, "Biometrics in the Mental Health Community," IEEE 11th International Conference for Upcoming Engineers (ICUE-2012), Ryerson University, Toronto, Canada, vol., no., pp., 2-3 Aug. 2012

Patents:

- US Patent: Biometric Matching Technology
  Abhinav Tiwari, Sanjoy Paul
  Publication Number: US 2013/0266193 A1
  Publication Date: Oct. 10, 2013

- US Patent: Biometric Authentication Technology

Abhinav Tiwari, Sanjoy Paul
Publication Number: US 2012/0314911 A1
Publication Date: Dec. 13, 2012

- European Patent: Biometric Matching Technology
  Abhinav Tiwari, Sanjoy Paul
  Publication Number: EP 2650822 A1
  Publication Date: Oct. 10, 2013

- European Patent: Biometric Authentication Technology
  Abhinav Tiwari, Sanjoy Paul
  Publication Number: EP 2533171 A2
  Publication Date: Dec. 12, 2012

- Chinese Patent: Biometric Authentication Technology
  Abhinav Tiwari, Sanjoy Paul
  Publication Number: CN 102842042 A
  Publication Date: Dec. 26, 2012