AUTHENTICATING SENSITIVE DIACRITICAL TEXTS USING RESIDUAL, DATA REPRESENTATION AND PATTERN MATCHING METHODS

SAQIB IQBAL HAKAK

FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2018

AUTHENTICATING SENSITIVE DIACRITICAL TEXTS USING RESIDUAL, DATA REPRESENTATION AND PATTERN MATCHING METHODS

SAQIB IQBAL HAKAK

THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2018

UNIVERSITY OF MALAYA
ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: SAQIB IQBAL HAKAK

Matric No: WHA150013

Name of Degree: DOCTOR OF PHILOSOPHY

Title of Thesis: AUTHENTICATING SENSITIVE DIACRITICAL TEXTS USING RESIDUAL, DATA REPRESENTATION AND PATTERN MATCHING METHODS.

Field of Study: Information Security

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date:

Subscribed and solemnly declared before,

Witness's Signature Date:

Designation: Senior Lecturer

Witness's Signature Date:

Designation: Associate Professor

AUTHENTICATING SENSITIVE DIACRITICAL TEXTS USING RESIDUAL, DATA REPRESENTATION AND PATTERN MATCHING METHODS

ABSTRACT

Diacritics play an important role in interpreting the meaning of a sentence through the proper pronunciation. Any text that needs diacritics is sensitive as any disarrangement of diacritics (intentional or unintentional) will result in complete misinterpretation of the text. There are different diacritics like punctuation symbols, extended letters (e.g. kashidas) and other symbols, that can be easily tampered to alter the original meaning of the text. There are limited studies focused on the authentication of such sensitive diacritical content (SDC). Most of the studies have removed the diacritics for authentication making the process questionable. Besides, the proliferation of such a sensitive content in different languages and formats on the internet has further exaggerated the issue of authentication involving search and retrieval phases. To address the mentioned issues, this thesis presents the different methods to authenticate the SDC with the aim to improve the searching and retrieval phases. The first method is based on the residual approach that authenticates any two similar sample texts written in different styles using one common database. It minimizes the overhead associated with maintaining the multiple databases. The objective is achieved using logical operations and the character segmentation. The second method is based on the representation of the diacritical text within the database to improve the retrieval performance for authentication of a single sentence (verse). The objective is achieved by creating individual nodes based on the total number of characters and placing each diacritical verse within its respective node. The last method is based on the pattern matching approach, where given multiple pattern input is authenticated from a given text. The purpose of exploring pattern matching approach is to authenticate multiple diacritical verses with improved time and space efficiency. The proposed method works by splitting the given pattern into two halves and searching for the respective halves. The searching of halves is achieved through two different algorithms based on the split approach and the parallel approach respectively. To show the practicality of the proposed methods, they are tested on sensitive diacritical text, which includes the Arabic Digital Holy Quran (DHQ). The reason for selecting the DHQ for evaluation purposes is its availability in different styles like uthmani and plain Arabic style that makes evaluation possible based on our first method. The second reason is the complexity of diacritics within DHQ and encoding scheme that decreases the authentication performance due to inefficient data representation and search/retrieval strategies. The mentioned reason made the evaluation of the second proposed method feasible and practical. Finally, for evaluating the pattern matching based approach, different sensitive texts including Arabic, French. Italian, English and Chinese were taken. The findings show that the first method manages to convert Uthmani and Plain Quranic verses into one common style with an accuracy of about 87 %. Similarly, the second method manages to authenticate single DHQ verse with the improvement in search time by approximately 70 % over the existing methods. Finally, the final method successfully authenticates multiple verses of different sensitive diacritical texts with improved computational time and memory consumption.

Keywords: Sensitive online content, Quran authentication, content integrity, information retrieval, exact matching.

PENGESAHAN TEKS DIAKRITIKAL YANG SENSITIF MENGGUNAKAN KAEDAH BERBAKI, PERWAKILAN DATA DAN KAEDAH PADANAN CORAK

ABSTRAK

Diakritik memainkan peranan penting dalam menafsirkan makna ayat melalui sebutan yang betul. Sebarang teks yang memerlukan tanda-tanda diakritik adalah sensitif kerana sebarang perubahan diakritik (sengaja atau tidak sengaja) akan menghasilkan salah tafsir terhadap teks yang lengkap. Terdapat diakritik yang berbeza seperti simbol tanda baca, huruf lanjutan (contohnya *kashidas*) dan simbol-simbol yang boleh diubahsuai dengan mudah untuk mengubah makna asal teks. Kajian yang memberi tumpuan kepada pengsahihan kandungan diakritik sensitif (KDS) masih terhad. Kebanyakan kajian telah mengeluarkan diakritik untuk tujuan pengsahihan menjadikan proses tersebut diragui kesahihannya. Selain itu, pertambahan pesat kandungan sensitif sedemikian dalam bahasa dan format yang berbeza-beza di internet telah memburukkan lagi isu pengsahihan yang melibatkan fasa carian dan fasa dapatan semula *(retrieval)*. Untuk menangani isu-isu tersebut, tesis ini menunjukkan pelbagai kaedah untuk mengsahihkan KDS dengan matlamat untuk memperbaiki fasa pencarian dan fasa dapatan semula *(retrieval)*. Kaedah pertama adalah berdasarkan pendekatan berbaki (*residual)* yang mengsahihkan dua contoh teks yang sama yang telah ditulis dalam gaya-gaya yang berbeza menggunakan satu pangkalan data sepunya. Ini meminimumkan overhed yang dikaitkan dengan pengekalan pangkalan data yang banyak. Objektif ini dicapai dengan menggunakan operasi logik dan segmentasi karakter (*character segmentation*). Kaedah kedua adalah berdasarkan perwakilan teks diakritik di dalam pangkalan data untuk meningkatkan prestasi dapatan semula(*retrieval*) untuk pengsahihan satu rangkap ayat (*verse*). Objektif ini dicapai dengan menghasilkan nod individu berdasarkan jumlah bilangan aksara dan meletakkan setiap ayat diakritikal dalam nod masing-masing. Kaedah terakhir adalah berdasarkan pendekatan padanan corak (*pattern matching*), di mana diberi banyak input corak disahkan dari teks tertentu. Tujuan meneroka pendekatan padanan corak adalah untuk mengsahihkan beberapa rangkap ayat diakritik dengan penggunaan masa dan ruang yang lebih efisien. Kaedah yang dicadangkan berfungsi untuk memisahkan corak yang diberikan kepada dua bahagian dan mencari separuh bahagian yang lain. Pencarian bahagian dicapai melalui penggunaan dua algoritma yang berbeza berdasarkan pendekatan belahan (*split*) dan pendekatan selari. Untuk memperlihatkan sejauh mana praktikaliti kaedah-kaedah yang dicadangkan, ujian dilaksanakan pada teks diakritik sensitif, termasuk Al-Quran Digital (AQD). Sebab pemilihan AQD untuk tujuan penilaian ialah ketersediaannya dalam gaya yang berbeza seperti Uthmani dan bahasa Arab biasa yang menjadikan penilaian mungkin berdasarkan kaedah pertama. Alasan kedua adalah kerumitan diakritik dalam skema pengkodan AQD dan skema pengekodan yang mengurangkan prestasi pengsahihan yang disebabkan oleh perwakilan data yang tidak cekap dan strategi pencarian/dapatan semula. Alasan tersebut menyebabkan penilaian kaedah kedua yang dicadangkan praktikal dan boleh dilaksanakan. Akhir sekali, untuk menilai pendekatan berdasarkan padanan corak, pelbagai teks sensitif termasuk bahasa Arab, Perancis, Itali, Inggeris, dan Cina telah diambil. Dapatan menunjukkan kaedah pertama berjaya mengubah ayat-ayat Al-Quran gaya Uthmani dan gaya biasa kepada satu gaya umum dengan ketepatan kira-kira 87%. Selain itu, kaedah kedua berjaya mengsahihkan rangkap ayat AQD tunggal dengan peningkatan dalam masa pencarian melebihi 70% berbanding dengan kaedah sedia ada. Akhir sekali, kaedah terakhir berjaya mengsahihkan beberapa rangkap teks dengan teks diakritikal yang berbeza dengan masa pengiraan dan penggunaan ingatan yang lebih baik.

Kata kunci: Kandungan sensitif dalam talian, Pengesahan Quran, Integriti kandungan, Dapatan kembali maklumat, pemadanan tepat.

ACKNOWLEDGEMENTS

All praises are due to Allah, the most beneficent and merciful, who is the Lord of the whole Universe and who created mankind from dust. There are a lot of people whom I would like to acknowledge in guiding me through this research work and attaining the doctorate degree.

First and foremost, I would like to thank and express the feeling of happiness to my academic supervisor: Dr.Amirrudin Kamsin, who guided me at every step and supported me each time when I needed his help. I am greatly indebted to him for all his guidance, support and concern both academically and otherwise. Secondly, I am short of expressions to show my gratitude to my co-supervisors: Associate Prof. (Dr). Mohd. Yamani Idna Idris and Associate Professor Dr.Omar Tayan (from Taibah University, KSA) for their efforts in providing me with valuable technical guidance to carry out required experiments. Thirdly, I am thankful to Dr. Shivakumara Palaiahnakote for his never-ending support in grasping me some strong concepts in text processing and journal writing.

At last, I would love to acknowledge my family including my wife Ms. Gulshan Amin, my mother Ms. Afroza, my brother Mr. Mohsin Iqbal Hakak, my Father (in Law) Mr. Mohammad Amin, my sister Ms. Samina Manzoor, my niece Mahira Mohsin Hakak who always supported me and kept me motivated to complete my doctorate degree. Last, but not the least, my love and token of thanks to my lab mates including Mr. Saber, Mr. Mujhtaba, Dr. Umawathy, Ms.Shamimi and Ms. Zarwina for their care and support.

Besides, I would love to express my token of thanks to funding agencies who sponsored my Doctorate studies including UMRG- RP003A-14HNE (the University of Malaya under the guidance of Professor (Dr. Abdullah Gani) and FRGS-FP003-2016 (the University of Malaya under the guidance of Dr. Amirrudin Kamsin)

I am grateful to Allah and I thank you all.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

AES	:	Advanced Encryption Standard
BDNM	:	Backward non-deterministic matching algorithm
BM	:	Boyer-Moore
BMT	:	Turbo Boyer Moore
CA	:	Certifying Authority
DCT	:	Discrete Cosine Transform
DHQ	:	Digital Holy Quran
DNA	:	Deoxyribonucleic Acid
DWT	:	Discrete Wavelet Domain
HMAC	:	Hash Message Authentication Code
JPEG	:	Joint Photographic Experts Group
LSB	:	Least significant bit
MD	:	Message Digest
PSNR	:	Peak signal-to-noise ratio
QQV	:	Quran quote verification
RA	:	Registration authority
SDC	:	Sensitive diacritical content
SHA	:	Secure Hash
SSM	:	Simple string matching
SST	:	Sum of square total
SVD	:	Singular Value Decomposition
SVM	:	Support Vector Machine
UTF	:	Unicode Transformation Format

CHAPTER 1: INTRODUCTION

This chapter contains the background for this study, highlights the research problem and motive for conducting this research, and formulates the problem statement and research questions. It includes the scope limitation of the research along with the research methodology.

1.1 Introduction

The authenticity/authentication is a set of policies and procedures that are necessary to validate the given input (Moukdad, 2013; Pinkerton, 2000). Two main procedures for authenticating digital content include retrieval/searching phase and the verified ground-truth (database). Without proper retrieval/search strategy, the authentication process will consume more time to process the given input (that can be any digital media)(Pinkerton, 2000). Currently, the issue of authentication is on rising due to the advent of modern gadgets. For the past few years, it is observed that the use of digital media uploaded and downloaded on the internet is steadily increasing. This steady increase of digital media over the internet is one of the major challenge faced by the researchers today in terms of determining the authenticity. The other issues arise due to the availability of the digital media in different formats like image, text, audio, and video. This availability of digital media in different formats makes identification of proper approaches to authenticate a particular format more challenging. Besides, the use of digital content has dramatically increased cases of copyright violations that spur researchers to study the issues related to the integrity, the authenticity of digital content and data vulnerability (Hakak et al., 2017a; Pan et al., 2004). This is the reason that the substantial amount of research is being undertaken in the area of data integrity, authentication and security. The data related to research in digital content authentication are shown in Figure 1.1.

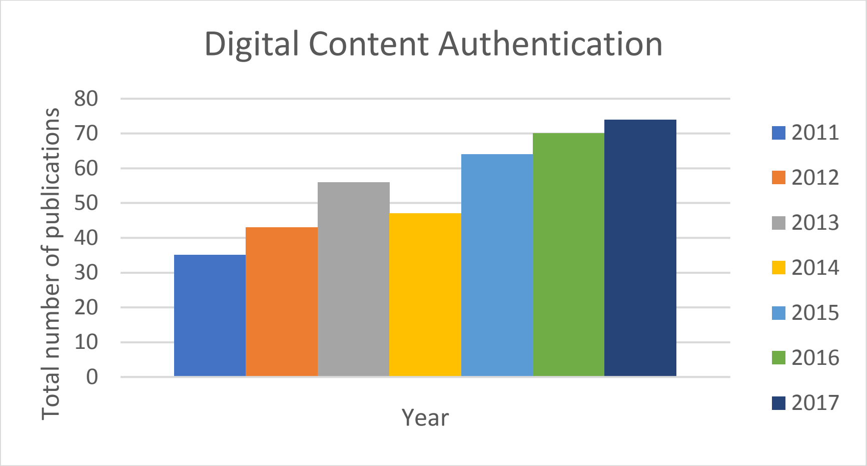


Figure 1.1: Research in the field of authentication/data integrity of online sensitive data (WoS, 2018)

The excessive reliance on the internet and the increase in users have exaggerated the problem of integrity and authenticity. According to the information made available by World Internet Statistics (Internet World Stats, 2018), the number of internet users has increased fivefold (as shown in Figure 1.2). Given this alarming trend, the rate of publishing sensitive digital content online is necessarily also on the rise. A lot of sensitive digital diacritical content is available online which can be accessed and downloaded from different sources, such as religious websites, social media websites and other online blogs.

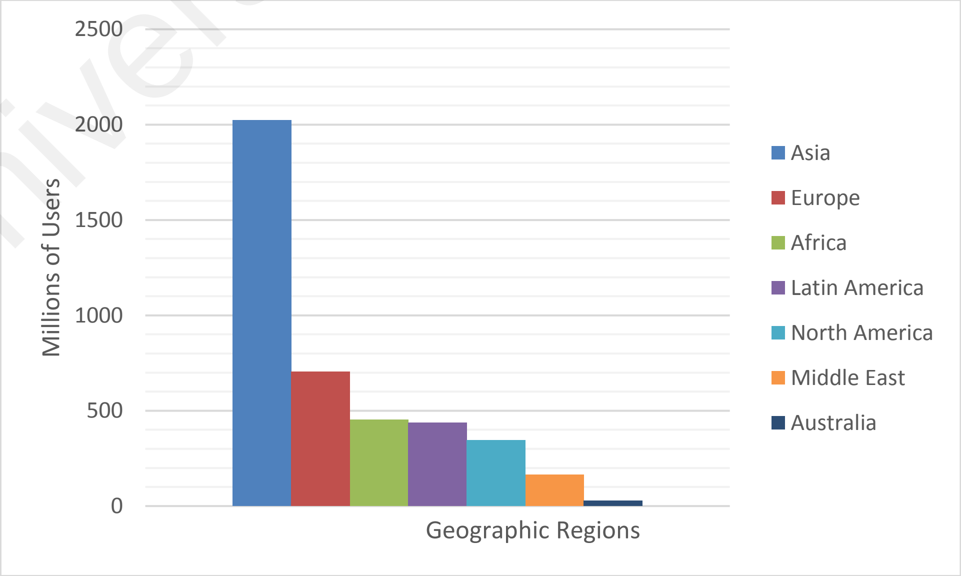


Figure 1.2: Number of Internet Users per year (Internet World Stats, 2018)

The research in authenticating sensitive diacritical text is quite new and emerging. Sensitive diacritical content (SDC) refers to the content that constitutes the material of utmost importance and requires the protection of confidentiality, integrity or availability and requires additional symbols (diacritics) for reading (Hakak et al., 2017b). This sensitive content may appear in the form of text, image, audio or video requiring different methods of authentication.

From the state-of-art, it is observed that very limited work is being pursued in this area. There is no existing study where the methods that can be used for determining the authenticity of diacritical text has been identified. Few studies that have tried to address the similar issues has focused on removal of diacritics that makes authentication process futile. Similarly, the other works have explored hashing process that involves the overhead of hash collision and inability to authenticate indistinguishable content using single database (Almazrooie et al., 2018; Alsmadi & Zarour, 2015; Hakak et al., 2018a). At last, the linear/binary search algorithms have been used along with regular expressions to authenticate sensitive diacritical text (Albujasim, 2014; Alginahi et al., 2013; Alshareef & Saddik, 2012). However, these approaches effect the retrieval efficiency and result in poor retrieval performance. Hence, there is potential to develop the approaches that can address the challenges of authenticating sensitive diacritical text.

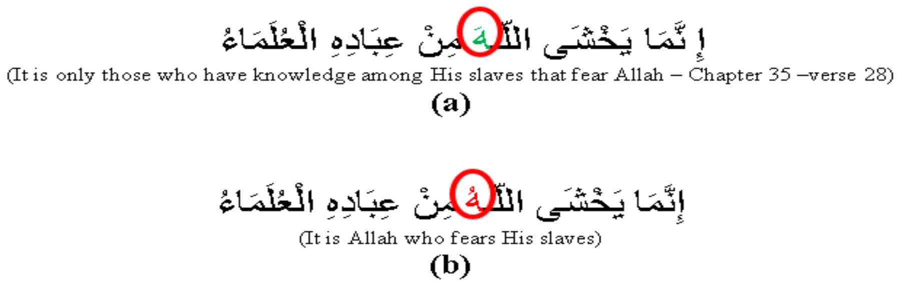
Some potential approaches that can help in addressing the issues in authenticating sensitive diacritical text include the use of logical operations (Wong, 2016). The use of logical operations can help in identifying the differences between the different writing styles of the same content. From the differences, substitution approach can be utilized to make different styles verifiable using one common database. Besides, there is a need to evaluate the effect of certain factors like the arrangement of data within the database to enhance the authentication mechanism. This can be achieved by focusing on data representation of the verified content. This approach can result in massive improvement of the overall retrieval performance for short/single verses. For multiple verse authentication, pattern matching approaches can be explored. These algorithms usually work by searching a given pattern from a given text using the skip-based method. The skip-based method includes the number of characters that can be skipped to save time and match the given input for authentication purposes. One of the standard pattern matching algorithms that are being benchmarked even after 20 years includes Boyer Moore algorithm (BM) (Boyer & Moore, 1977). The reason of including BM algorithm in current studies is due to the nature of the pattern within the text. This algorithm is still best for long patterns due to its efficient shift process (Rahim et al., 2017; Sri et al., 2018). All the mentioned approaches can be used to authenticate the sensitive diacritical text. As there is massive sensitive content available, it is a tedious task to evaluate the practicability of the above-mentioned solutions. Hence, we include the case study of Digital Quran (DHQ) for evaluation purposes due to its availability in different styles and complex diacritics.

Digital copies of the text of the Holy Quran (DHQ), which constitutes the most authentic and unaltered religious text of all times, falls into the category of highly sensitive online content with respect to tampering. It constitutes every Muslim’s duty to protect the authenticity and integrity of the Quranic text and message (Tayan et al., 2014). The Quranic content is available in the form of simple text (binary format) or images on numerous websites in different Arabic script styles. For example, numerous symbols and diacritics constitute an integral part of the Quranic text, and the modification of just one symbol may change the meaning of a whole sentence or verse. If one single verse is misunderstood or misinterpreted, it creates a lot of confusion in the minds of its Muslim readers (Alsmadi & Zarour, 2015). Thus, different methods are needed that can help authenticate the sensitive diacritical texts like the digital copies or verses of the Holy Quran available online, which come in various different Arabic script styles.

1.2 Motivation

The research area of analyzing sensitive online content is quite new and growing. Thus, it requires significant research efforts in terms of identifying the optimal methods used to assess and evaluate their level of performance and efficiency in connection to different formats such as text and image. The massive increase in the use of digital content over the last few years, and issues of copyright, authenticity, and integrity of digital content and data vulnerability constitute the main motivating factors for carrying on the research in this area. Given the enormous amount of digital content published on the internet, the case of the Holy Quran is considered

DHQ is represented in the form of different script styles, the most established being Uthmani (Sabbah & Selamat, 2013) and plain Quranic style (most commonly used in Asia). Uthmani script involves more complex diacritics compared to the plain script. In both cases, diacritics play a vital role in determining the semantics or linguistic meaning the text reflects. For example, in Figure 1.3(a), an authentic Quranic verse with proper arrangement of diacritics is given with proper translation. However, in Figure 1.3 (b), the same verse has been tampered with by changing just one diacritic (highlighted in red) resulting in a completely opposite meaning.



إِنَّمَا يَخْشَى اللَّهَ مِنْ عِبَادِهِ الْعُلَمَاءُ

(It is only those who have knowledge among His slaves that fear Allah – Chapter 35 –verse 28)

(a)

إِنَّمَا يَخْشَى اللَّهُ مِنْ عِبَادِهِ الْعُلَمَاءُ

(It is Allah who fears His slaves)

(b)

Figure 1.3: (a)Authentic verse (b) Tampered verse

The two examples illustrated above show the impact of diacritics on the authenticity and correct representation of the Quranic verses. Diacritics affect the meaning and have a significant impact on retrieval. Previous studies have shown that the complexity of diacritics results in poor retrieval of output with increased time complexity (Darwish & Magdy, 2014). All the mentioned factors and the significance of DHQ further strengthened the motivation to proceed with the research.

1.3 Problem Statement

Diacritics are the small marks written above or beneath the letters necessary for the correct recitation and grammatical understanding of the text. The availability of indistinguishable sensitive diacritical text in different styles makes the authenticating process very complicated due to the requirement of the ground truth (reference database) for each text sample. Besides, diacritical content like Arabic text requires more than 7 bits to represent a single character, this results in poor retrieval performance and increases time complexity. This is the reason, most of the existing sensitive diacritical text authentication approaches like in case of DHQ either fail to retrieve the required input or take a considerable amount of time for processing the request. The final issue arises involving authentication of multiple complex diacritical verses that incurs more time compared to the plain single verse due to the processing of more bits. To address the above-mentioned challenges, require prior authentication methods by taking DHQ as a case study.

1.4 Research Questions (RQs)

This study addressed the following research questions (RQs) that were formulated corresponding to the research objectives to delineate the scope of this research:

a) What are the issues with the state-of-the-art in authenticating the sensitive diacritical text?

b) How logical operations and character segmentation can be integrated to authenticate different styles of diacritical text?

c) Can the efficient data representation improve the retrieval process and enhance the authentication process of diacritical text?

d) Can a multi-lingual algorithm be developed for the authentication of multiple verses of sensitive diacritical text?

1.5 Research Objectives

The aim of this thesis is to determine the authenticity of sensitive diacritical texts by conducting a case study of digital Quran texts. To achieve this purpose, we have formulated the following objectives:

- To investigate the state-of-art issues in authenticating (searching and retrieving) the sensitive diacritical texts;
- To propose and evaluate a method for converting different script styles of sensitive diacritical texts into a standard script style;
- To propose and evaluate a method for diacritical text storage and retrieval;
- To propose and evaluate a method for determining the authenticity of multiple diacritical texts with improved time complexity.

1.6 Research Significance

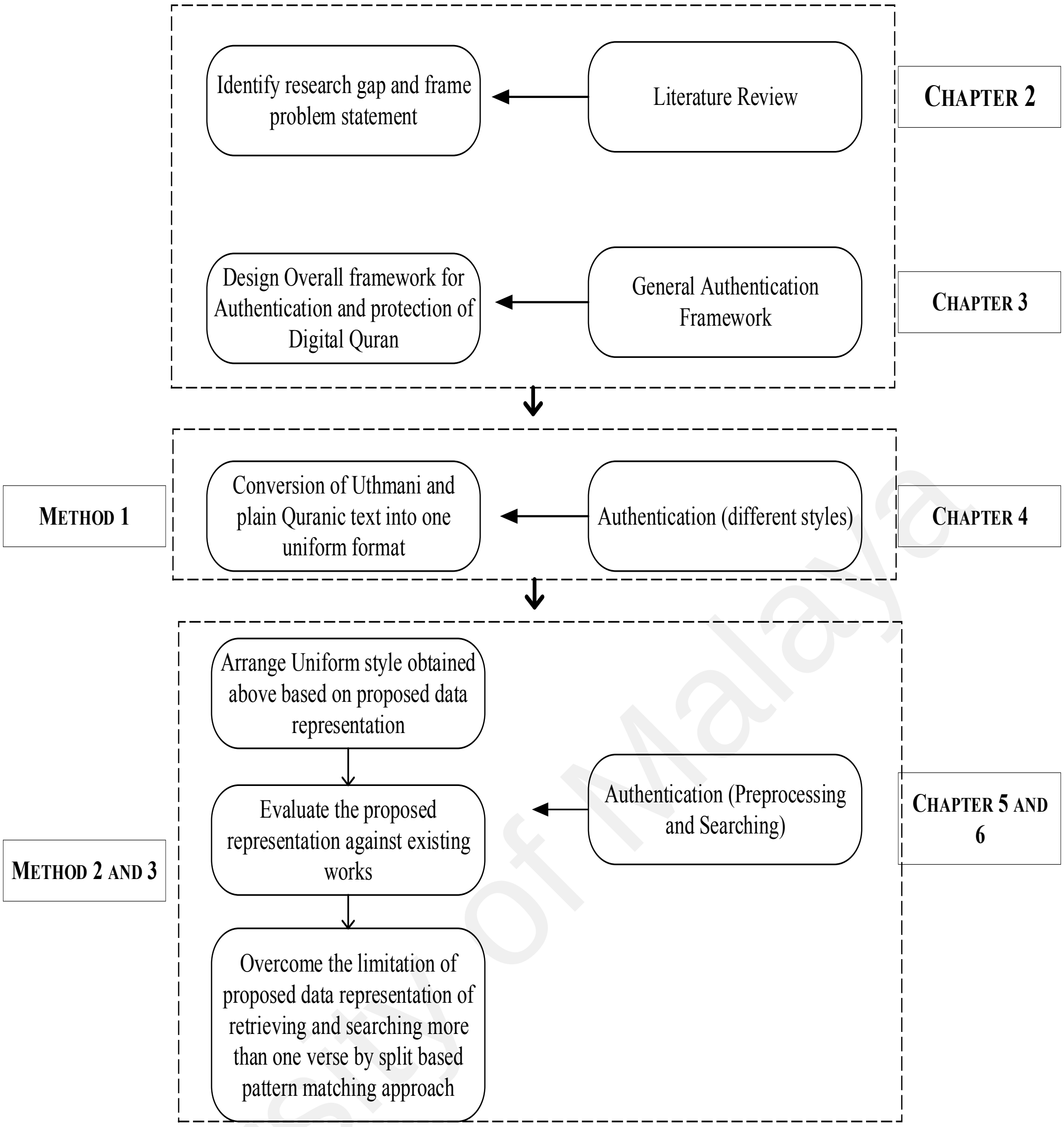
The aim of this research is to propose different methods for authenticating diacritical sensitive texts like Quranic verses available online. Such an authentication method would constitute part of the worldwide Muslim effort to preserve and protect the correct and original text and meaning of DHQ. The authentication methods will also assist Muslims and non-Muslims in the study of the Quran online without having to worry about being exposed to the fraudulent and misleading content. Furthermore, the proposed taxonomies related to online sensitive content and other open issues point out possible directions to be taken in future research. The present research proposes a novel way of authenticating more than one script style of DHQ using a single common database. This is hoped to motivate researchers to explore other methods that can further improve the accuracy of conversion and include other types of script styles. The methods proposed in the current research can be explored further in order to test on other kinds of sensitive diacritical texts.

1.7 Research Methodology

The research project is divided into two study phases: Phase 1 consists of a literature review and phase 2 relates to authentication (comprising of preprocessing and search phases)

The first phase of the research covers the literature review, which introduces the basic concepts and issues pertaining to the scope of study covering the first objective. Related work related to preserving the content integrity of sensitive diacritical texts with the case study of DHQ is studied and new taxonomies that can pave future research directions are presented. To encounter the second objective, the residual method is proposed. For authenticating phase that involves retrieving and searching Quranic verses, a new data representation is proposed. A fourth and final method is proposed based on a pattern matching approach to improve the time efficiency of multilingual text.

The overall research methodology is discussed in Chapter 3 and the individual research methodology of the methods mentioned above is described and presented individually in order. Chapters 4, 5 and 6 are organized into an introduction, related works, research methodology, results, and conclusion. The research flow and its corresponding research methods are summarized in Figure 1.4.



Identify research gap and frame problem statement
Literature Review
CHAPTER 2
Design Overall framework for Authentication and protection of Digital Quran
General Authentication Framework
CHAPTER 3
METHOD 1
Conversion of Uthmani and plain Quranic text into one uniform format
Authentication (different styles)
CHAPTER 4
Arrange Uniform style obtained above based on proposed data representation
Evaluate the proposed representation against existing works
Overcome the limitation of proposed data representation of retrieving and searching more than one verse by split based pattern matching approach
Authentication (Preprocessing and Searching)
METHOD 2 AND 3
CHAPTER 5 AND 6

Figure 1.4: Research flow and its corresponding research methods proposed.

1.8 Contribution

The contribution of this research includes:

1. Identification of approaches that can be used for preserving the content integrity of sensitive diacritical text through taxonomies;
2. Framework for authenticating and protecting sensitive diacritical text;
3. Method for authenticating the different script styles of using single database;
4. Efficient data representation method for authenticating single diacritical verses;

5. New split-based searching algorithms for authenticating multiple diacritical verses;
6. Recommendations for future research in the area of sensitive diacritical text.

The workability of the proposed technical approaches has been shown by taking the case study of the DHQ. This study also constitutes the first survey carried out on the different approaches used for preserving the integrity of the DHQ;

1.9 Thesis Organization

Chapter 2 provides an extensive overview of the various types of SDC. A general introduction of techniques used to authenticate SDC is given. The sensitive nature of DHQ and important diacritics along with related studies are reviewed together with their strengths and limitations. The research methodology is briefly summarized in Chapter 3. Chapter 4 presents the limitations of the existing approaches involved in determining the authenticity of SDC. It describes the proposed residual based approach to convert different styles of sensitive diacritical text using a single database. The proposed approach is evaluated and benchmarked against the existing approaches by taking the case study of DHQ. Chapter 5 explains the limitations of the existing diacritical search engines like Quranic engines. A selection of related studies is consulted in view of their limitations. An efficient data representation approach for the retrieval of single diacritical text verse is proposed that improves the overall retrieval performance. The evaluation has been done using DHQ verses and compared to the existing approaches. The limitations of the proposed work and previous works are also discussed. Chapter 6 presents the method deemed most suitable to overcome the limitations of the method proposed in chapter 5, i.e. authenticating single diacritical verses. The proposed method is based on the concept of string or pattern matching algorithms, which are briefly explained. The proposed algorithm is evaluated on different datasets of English, Arabic, Italian, French and Chinese. The limitations of the existing studies and the proposed approach are briefly highlighted. Finally, chapter 7 contains the conclusion and a brief outline of future research.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter discusses previous related works that aimed at authenticating and protecting one of the most sensitive diacritical content i.e. the DHQ. The Quranic content can be divided into three categories: image, text and audio/video-based content. Due to the vast amount of digital content available on the Internet, this study has focused only on text-based approaches along with their limitations. The basic approaches adopted by researchers to authenticate and protect text and image-based content is briefly explained alongside their advantages and limitations. Finally, open research issues and conclusion are drawn.

2.2 Classification of Sensitive Online Content Based on Format

Sensitive online content can be defined as content whose accuracy, security and correctness are of crucial and critical importance to users (Computer Security Act 1987)(Hakak et al., 2017b). Sensitive online content can be classified into four different types based on their respective format (as shown in Figure 2.1). Such a classification enables researchers to identify and analyze different techniques for protecting and verifying the integrity of the sensitive online content.

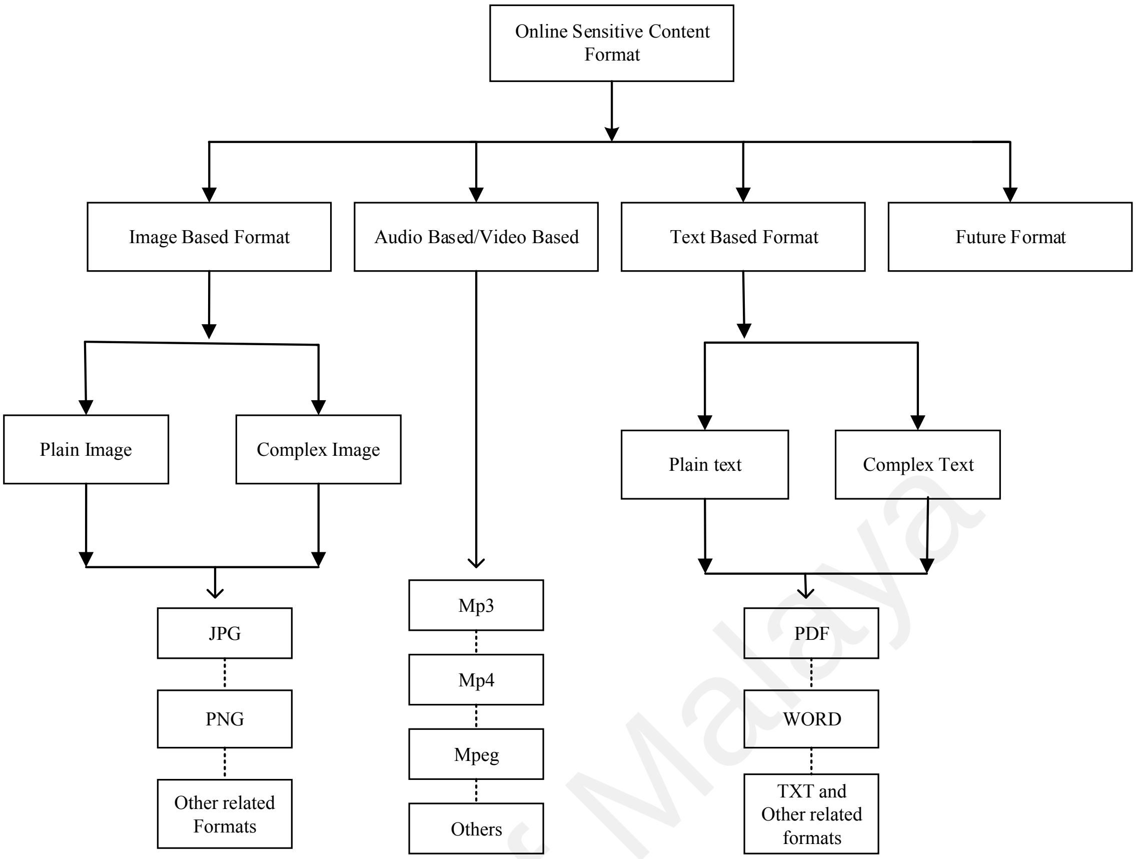
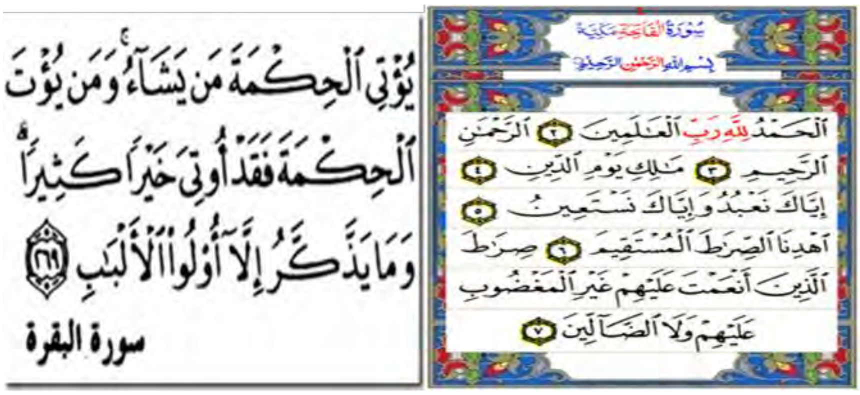


Figure 2.1: Classification of sensitive online content based on the format

a) Image-Based Format: The Internet contains plentiful image-based content that is readily accessible while surfing different websites. For example, books are presented in image format. We have categorized 'image' into two sub-categories, plain and complex images. Plain images include simple pictures with average color details and clarity, whereas complex images include pictures with high color details with not much clarity. The two types of images are shown in Figure 2.2. Both plain, as well as complex images, can be displayed in different formats like JPEG, TIF, and GIF. This area falls under image processing, and there are different kinds of techniques used for checking their integrity.



يُؤْتِي ٱلْحِكْمَةَ مَن يَشَآءُ ۚ وَمَن يُؤْتَ
ٱلْحِكْمَةَ فَقَدْ أُوتِيَ خَيْرًا كَثِيرًا ۗ
وَمَا يَذَّكَّرُ إِلَّآ أُو۟لُوا۟ ٱلْأَلْبَٰبِ ٢٦٩
سورة البقرة

سُورَةُ الفَاتِحَةِ مَكِّيَّة
بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ
ٱلْحَمْدُ لِلَّهِ رَبِّ ٱلْعَٰلَمِينَ ٢ ٱلرَّحْمَٰنِ
ٱلرَّحِيمِ ٣ مَٰلِكِ يَوْمِ ٱلدِّينِ ٤
إِيَّاكَ نَعْبُدُ وَإِيَّاكَ نَسْتَعِينُ ٥
ٱهْدِنَا ٱلصِّرَٰطَ ٱلْمُسْتَقِيمَ ٦ صِرَٰطَ
ٱلَّذِينَ أَنْعَمْتَ عَلَيْهِمْ غَيْرِ ٱلْمَغْضُوبِ
عَلَيْهِمْ وَلَا ٱلضَّآلِّينَ ٧

(a) (b)

Figure 2.2: Quranic Images (a) Plain, and (b) complex

b) Audio/Video-Based Format: A huge amount of content is available online in the form of audio and video formats. The most popular formats include mp3, mk3, MPEG, and mp4. There are different techniques that can be applied in order to check the authenticity of this content.

c) Text-based Format: One of the most suitable and available online contents is in text format, plain text, and complex text. The plain text refers to simple English which uses the ASCII format while complex text includes any text which takes more than one byte and uses different encoding techniques like UTF 8 and 16 as shown in Figure 2.3. Complex text may include Arabic text, Persian text, text with symbols and so on (Arslan, 2015). The different formats available are for example word format and pdf format. As compared to the image-based format, research done on the text-based work has been limited. This may be due to the availability of binary texts in different styles that need proper authentication mechanisms.

Hello ٱلْحَمْدُ لِلَّهِ رَبِّ ٱلْعَٰلَمِينَ

(a) (b)

Figure 2.3: (a) Plain text, and (b) Complex Arabic text

d) Future Formats: We believe that in the near future, a universal format is going to be invented, which is going to work for all types of media like images, text, video and so on. To make our taxonomy scalable, a future format is included for further refinement of the proposed taxonomy (Hakak et al., 2017b).

We have discussed different resource formats available online. This classification can be useful for researchers in categorizing techniques specifically based on formats, and in determining and assessing their specific limitations and advantages. For text and image-based format, there are different approaches that are being employed to determine the authenticity and content integrity of DHQ.

2.3 Approaches to determine authenticity and preserving the content integrity of Quranic Content

Content integrity refers to the process of authenticating or verifying whether a given content is in its original form or not. Thus, in this section, all basic approaches used for content integrity purposes with respect to Quranic content are discussed. The summary of research done in the content integrity of Quranic content is discussed at the end of each respective approach for watermarking, cryptography, steganography, pattern-based and whole document-based approaches.

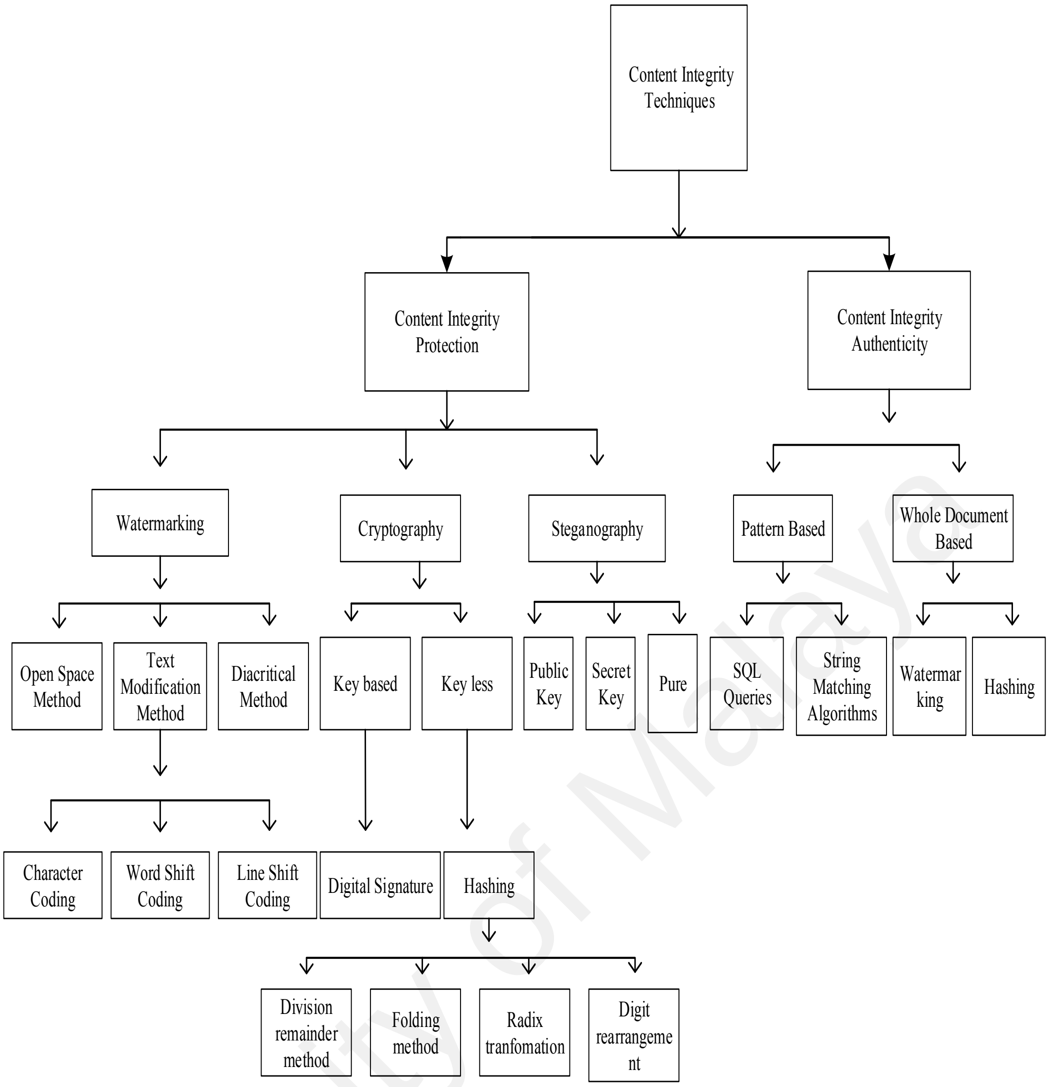


Figure 2.4: Taxonomy of Content Integrity Approaches

There are usually two concerns that the user has while reading sensitive Arabic texts like the Quran on the Internet. One primary concern is how to verify whether a certain verse/sentence is accurate and correct or not, and the other being the best way in which to protect the own uploaded Quranic content from tampering or any modification. Considering these two concerns, we have classified the content integrity approaches into two categories, namely content integrity protection (protection of DHQ) and content integrity authentication (verification of DHQ), as shown in Figure 2.4 above.

The approach for content integrity protection can be subdivided into the three categories: watermarking, cryptography and steganography. Finally, content integrity authentication can be classified into pattern-based and whole-document based approaches. All those approaches are briefly described as follows:

2.3.1 Content Integrity Protection

The main factors that any content integrity protection and verification mechanism must have are as follows (Daraee & Mozaffari, 2014; Haouzia & Noumeir, 2008; Tao et al., 2014):

- Imperceptibility: It means the property of being invisible. The document to be protected from tampering much have a security feature embedded within it such that it is not noticeable to the audience (Arnold et al., 2002; Makbol et al., 2016).
- Robustness: The secured approach must be robust enough to tolerate any kind of attacks (Arnold et al., 2002; Makbol et al., 2016; Nin & Ricciardi, 2013).
- Security: The approach used for preserving content integrity must be secure enough to satisfy the condition of robustness (Arnold et al., 2002).
- Computational cost: The approach used to preserve the content integrity of documents must be scalable in the future of future computers with less computational overhead (Arnold et al., 2002).

Based on these factors, different techniques can be used for data integrity protection. The following techniques have been used for protecting the image based Quranic content from possible tampering or modification.

2.3.1.1 Watermarking

Digital watermarking is used in protecting digital media from possible tampering. It can also be used for authentication purposes. In digital watermarking, a certain piece of information like a company logo or text is added to digital media (e.g. image and video) in order to secure the content and owner identification. It is used in many areas like broadcast monitoring, authentication, and copyright protection (Singh & Chadha, 2013). An efficient watermark should fulfill the following criteria: robust, imperceptible, secure, verifiable, good capacity, less complex, reliable (Brassil et al., 1999). Watermarking can be either in the form of image watermarking or text watermarking. In image watermarking, an image is used for processing based on the pixel or block approach. In text watermarking, the text is used to watermark the information. The methods that have been employed to protect digital Quran include:

a) Diacritical based methods: There are special symbols known as diacritics in the Arabic language that are used to read highly grammatical scriptures like the Arabic Quran. There are mainly eight different diacritical symbols that are used to hide binary bits and protect the document (Hakak et al., 2017c).

b) Open space methods: The protection is achieved by hiding the watermark in extra spaces. Those extra spaces can be found at the beginning, middle or end of the text. Such spaces are also known as white spaces (Bender et al., 1996).

c) Text modification methods: The text is modified physically. There are three main approaches that are employed to achieve the purpose of protecting sensitive Arabic content using text modification methods i.e. Line coding, word-shift coding, and character coding. Line coding (Hakak et al., 2017b): The text line is altered vertically, which is not visible to users. Thus, during the decoding process, line displacement helps to identify any tampered text (Brassil et al., 1999). Word-shift coding: This technique works in the horizontal direction. The location of the word is shifted horizontally within the text line. The unmoved words serve as reference locations during the decoding process (Brassil et al., 1999). Character coding: This is a feature-based technique where the features of a character like height or font are used for watermarking. The feature of a character is changed to embed a security code for verification purposes. The height of an individual character can be altered, or its position can be changed in relation to other characters. During the decoding process, some characters are left unchanged (Brassil et al., 1999). This technique constitutes one of the most widely used techniques in preserving content integrity (Khare, Shivakumara, & Raveendran, 2014).

A summary of watermark-based approaches for preserving the content integrity of digital Quranic content is given below:

The *Kashida*-based approach for protecting the digital copies of the Quran from tampering has been proposed (Gutub et al., 2010). The authors have used the feature coding technique of watermarking. In this approach, a security code is embedded with *kashida* to watermark the text. *Kashida* is a way of writing Arabic by elongating the length of the characters. The proposed method is suitable for copyright issues (Gutub et al., 2010).

The two crucial challenges while handling sensitive online content are content protection and copyright protection (Tayan et al., 2013). Tayan et al. 2013 have proposed the zero-watermarking approach for content verification and authentication. Authentication is achieved by embedding the documents with a specific data sequence obtained from the watermark logo. The specific key is then generated using logical XOR, yet the word size limit for which the key has to be generated is not mentioned. Even though the study calculates the key for a whole document similar to the hashing process. The calculated key is used by the certifying authority (CA) for authentication purposes during the decoding phase. The claim raised in support of this approach is that the Unicode values of all characters are calculated and added to produce a sum after which the parity bit is added and the key generated. (Tayan et al., 2013).

In another study by (AlAhmad et al., 2013a), the focus is on the authentication of Quranic images. The invisible watermarking technique based on LSB is used. In order to generate the watermark for the pdf file, a DCT algorithm is used to reduce the extraction time. For tamper detection, the hashing approach is used.

Taking the online content of the web into consideration, Abuhaija et al., 2013 (Abuhaija et al., 2013) proposed a framework named ITRUST. However, the implementation of the authentication process is not sufficiently elucidated. The authors only explain that the Registration Authority (RA) will be responsible for authenticating the content on the websites. However, the role of the watermarking logo provider remains unclear (Abuhaija et al., 2013).

The authors in (Syifak Izhar et al., 2013) have proposed another watermarking approach based on the SVD technique to authenticate Quranic images available online. In the SVD technique, the image is transformed into different matrices and each matrix processed. The advantage of this technique lies in its robustness and security against geometrical attacks. (Laouamer & Tayan, 2013).

In order to protect Quranic images, the research completed by (Kurniawan et al., 2013a) proposed a method whose focus is on protecting and then authenticating the watermarked Quranic image. The original Quranic image is hashed to obtain the initial authentication code. This authentication code is then encrypted using a private key to obtain the secured and authenticated code. This secured code (L) is used to counter local attacks and stored in binary format. Wavelet domain is used to embed this secure code with the host image using DWT transformation. The embedding process is done in coefficient wavelet (Ch) at resolution level 1-L. For authentication purposes, the inversion of the whole process is repeated. The authors have evaluated the proposed approach on four images of the Holy Quran. The experiments served the purpose of evaluating the localization capability of image tampering, fragility to JPEG compression under various quality factors (QF), and the quality of the image after the watermarking process. The proposed technique maintains a fine image quality.

Another study (Kamsin et al., 2014) discussed the authentication of the Quran in all possible formats. With the digitization and the rapid and steady increase in the numbers of internet users, a reliable and universal Quran authentication system is needed in order to detect fake verses speedily. The program to achieve the objective of developing such a Quran authentication system is also discussed.

(Kurniawan et al., 2014) This study discussed a method similar to the previous work published by the same author, except using a different evaluation parameter. Here, the image is also transformed into the wavelet domain using DWT. Using the block-based approach, the image is then divided into several blocks and each block is embedded with watermark bits for authentication purposes. Three parameters, i.e. PSNR, Pearson Correlation Coefficient (PCC) and Normalized Hamming Distance (NHD) are taken for evaluation purposes (Kurniawan et al., 2013b).

Tayan et. al (2014) proposed an approach verification of digital text document integrity. This effort is similar to the approach discussed above in (Tayan et al., 2013). The main difference lies in the fact that in (Tayan et al., 2013), the word size related to the key generation is not mentioned, whereas the word size limit is fixed at two in (Tayan et al., 2014).

In order to meet the specific requirement of verification online content, (Sabbah & Selamat, 2014) aimed to develop a machine learning model able to classify the online words into Quranic and non-Quranic words using a support vector machine. A classification model is applied to the words extracted from the online source. However, before extracting the words, all symbols, diacritics, and non-Arabic letters are removed in the filtering phase. The classification model is evaluated using three parameters, i.e. Accuracy, precision and F-measure (Sabbah & Selamat, 2014).

The main finding and limitations of the above-mentioned studies that have focused on the image-based format are shown in Table 2.1.

Table 2.1: Advantages and drawbacks of Watermarking approaches used in authenticating/protecting DHQ content

Authors	Approach used	Findings	Limitations	Evaluate Metrics used
(Gutub et al., 2010)	Feature code watermarking	A method based on embedding security bits in kashida is proposed.	1. Limited to image only. 2. Need to test the proposed approach using a different kind of attacks.	There is no specific metric, taken for evaluation purposes.
(Tayan et al., 2013)	Zero - watermarking	The Zero-watermarking approach has been identified as the most effective technique to authenticate image-based documents	Limited to image format only. The whole document is hashed which is similar to hashing	Not any evaluation parameter is taken. Only encoding time and decoding time mentioned.
(AlAhmad et al., 2013a)	Invisible watermarking and hashing	An invisible watermarking approach based on LSB is proposed.	1. Results are not valid results due to lack of experiments. 2. Limited to Pdf format only	No Experiments were done and discussed.
(Abuhaija et al., 2013)	Watermarking	A general framework regarding authentication of website content is proposed.	2. Based on the manual authentication mechanism where Registration authority is assigned to authenticate websites.	Not experimental. General architecture proposed.

Table 2.1: Continued

Authors	Approach used	Findings	Limitations	Evaluate Metrics used
(Syifak Izhar et al., 2013)	Fragile Watermarking	Image-based authentication of Quranic images based on fragile watermarking is proposed and the proposed algorithm shows significant improvement.	Limited to image only.	Top four Quran applications from Android have been taken for experimental purposes. Four evaluation parameters, i.e. Average processing time, Average PSNR value detection and recovery are taken.
(Khalil et al., 2014)	Fragile Watermarking	The Fragile watermarking approach has been proposed to protect the Quran from tampering.	Limited to image format only	Two metrics, i.e. Bit Error rate and PSNR are taken for evaluation purposes.
(Laouamer & Tayan, 2013)	Watermarking	An enhanced approach based on singular value decomposition(SVD) for watermarking the data is proposed	Limited to the image. There are no benchmark results shown from previous work.	For evaluation purposes 5 parameters have been taken: Peak Signal-to-Noise-Ratio (PSNR), Structural Similarity Index (Hassan et al.), Visual Information Fidelity (VIF), the Universal Quality Index, Noise Quality Measure (NQM)
(Tayan et al., 2014)	A hybrid approach of watermarking and digital signature	The Zero-watermarking approach has been identified as an effective technique to authenticate image-based documents	This approach is prone to fail in authenticating a particular line in a document. The approach considers a whole document file as an input. Overhead of storing keys.	No specific evaluation parameter is taken. Only encoding time and decoding time mentioned.
(Sabbah & Selamat, 2014)	Machine Learning Approach	The machine learning approach has been proposed for classifying Quranic words from Non-Quranic words.	Limited to image format only.	Three parameters, i.e. Accuracy, precision, and F-measure have been taken for evaluation purposes on 4 different datasets.

2.3.1.2 Cryptography

It is one of the most widely used techniques for securing data. There is plentiful sensitive online information in the form of credit cards, banking transactions and the like that needs to be protected. Cryptography involves converting the human-readable text into unreadable text (also known as ciphertext) for any unauthorized person through encryption, generation of keys and decryption. Encryption is the process of converting plain text into cipher text, and decryption is the reverse of encryption. Keys are used to unlock the encryption phase (Mitali & Sharma, 2014). The main advantages of cryptography are securing confidential information between sender and receiver, authentication for proof of identity, integrity check to ensure that the message has not been tampered with and non-repudiation, which does not allow the sender to claim that the message has not been sent.

Cryptography can be classified into two approaches: key-based and keyless (as shown in Figure 2.4). The key-based approach consists of symmetric cryptography and asymmetric key cryptography. In symmetric key cryptography, one key is used for encryption as well as decryption and in asymmetric key cryptography two keys are used. The most widely used algorithms in key-based cryptography include AES and DES algorithms. DES has a key size of 56 bits which is quite small for many applications available today. Due to the advancement in computing power, it is not advisable to use the approach. AES, on the other hand, which is considered as an improvement over the DES approach, is too complex compared to the size of the key length which is quite long. Blowfish is considered as both secure and flexible as compared to the above-mentioned approaches. It uses simple primitive operations like XOR, table-lookup and can execute in below 5KB memory. Asymmetric approaches, on the other hand, such as RSA, which uses public key cryptography are very slow in processing (Mitali & Sharma, 2014).

One of the most popular approaches to achieve key based cryptography is through the use of digital signatures. A Digital signature refers to a mathematical technique used to authenticate and validate the integrity of any message or digital document (Tayan et al., 2014). This technique is useful to detect any kind of tampering or impersonation in digital communications. In digital signatures, a one-way hash of electronic data is created. A private key is used to encrypt the hash. Thus, the combination of the hashing approach and encryption using the private key creates the digital signature. It consists of the following three algorithms. The *key generation* algorithm selects the private key and outputs the private key and the corresponding public key. The *signing* algorithm receives the message and private keys as input and produces a signature. The *signature verifying* algorithm checks the given the message, the public key and signature for authenticity (Schellenkens, 2004).

The second approach is keyless, which involves hashing and random number generation. Among the keyless approaches under cryptography, the most popular method is hashing. The following methods are used for hashing:

- *Division-remainder method:* It calculates the total size of a number of items. This number is then used as a divisor against the original key value to extract the quotient and remainder. The remainder after the division process is the hashing value.
- *Folding method:* It divides the original value into several sub-values which are then added. It uses the last four digits of this summation as the hash key.
- *Radix transformation method:* It changes the radix or number base. This change in radix results in a different sequence of digits which is used as the hash key.
- *Digit rearrangement method:* It takes and rearranges part of the original value or key, and then uses that sequence as the hash value or key.

2.3.1.3 Steganography

It is an information-hiding technique in which data is hidden in a cover media to make that information inaccessible to others (Sumathi et al., 2014). The difference between cryptography and steganography is that cryptography involves protection of the message content while steganography conceals the existence of the original message (Katzenbeisser & Petitcolas, 2000). The idea is to embed the original message in a cover and send it to the destination secretly. The stego key is used to restrict the detection or recovery of the original message (Bennett, 2004).

Steganography can be classified into three categories (Sumathi et al., 2014):

a) Pure Steganography: It does not require a stego key as no other party is aware of the communication.

b) Secret Key Steganography: The stego key is used for communication purposes. This type of technique is prone to interception

c) Public Key Steganography: Private and public keys are used for secure communication.

All the above approaches use the following methods (Sumathi et al., 2014):

I. Substitution methods: Redundant parts of a cover are substituted with a secret message.

II. Transform domain technique: Secret information is embedded in the transform space of the frequency domain.

III. Distortion technique: Information is stored using signal distortion, and the deviation from the original cover is measured in the decoding step.

IV. Statistical method: Information is encoded by changing several statistical properties of the cover and uses hypothesis testing in the extraction process (Sumathi et al., 2014).

The summary of cryptographic and stenographic-based approaches for preserving the content integrity of Digital Quran is as follows:

The ready availability of digital text formats on the internet in the form of websites, articles, e-books and the like make it easy to copy and tamper with it and forge a fake text substituting the original. Therefore, the authors (Gutub et al., 2010) proposes a steganography based approach whose main objective is to provide copyright protection and prevent illegal copying and diffusing. In the study, the stenographic technique is used to protect the text of the Quran from any subsequent modification. The secret bits are embedded in extensible characters of the Arabic script. The drawback of this kashida-based approach lies in the elongation of the letter characters which takes up more space, requires more bytes and thus occupies more memory in order to authenticate the Quranic images. The research also proposes the use of the fragile watermarking technique. The image is divided into blocks and each block is numbered in a spiral manner, starting from the center resembling a ring form. The numbering is done so that the watermarked blocks are relocated at a minimum distance away from the original blocks. All blocks are then mapped using an equation $B = [(k * s) \bmod Nb] + 1$, *B* is watermarked block, *s* is spiral, *Nb* is the block numbers, and *k* is the secret key which is the highest prime number from the result of block numbers divided by 2. In the second phase, the average intensity of each block is calculated by setting the LSB of each pixel within the block zero.

To block any attempt to distort the text of the Holy Quran such as Galan application, Alshaikhli et al. (2013) combined the two well-known approaches of AES and RSA resulting in a hybrid approach. Its focus is on protecting the Holy Quran from any alteration. However, there is no clear methodology mentioned on how this protection can be achieved. There is no mention of a newly proposed algorithm and the result is not duly evaluated (AlAhmad et al., 2013b).

A model has been proposed for the authentication of Quranic verses. The author claims that document control and digital signature are the two most widely used approaches in authenticating documents. Document control is giving permission before and after publishing the document online. In the digital signature, signed documents are verified by the individual signing it. The study focuses on integrity checking. The author mentions that it is a challenge to read or place the Arabic diacritics correctly. The hash approach is used to calculate the particular verse, the hash value is then compared with the hash value in the database. The major drawback of this approach is that only one single verse can be checked at a time. Different verses are tested using different hashing approaches.

The summary of works based on cryptography and steganography approaches to preserve the content integrity of DHQ is shown in Table 2.2 respectively.

Table 2.2: Advantages and drawbacks of Cryptographic and Stenographic approaches used in authenticating/protecting DHQ content

Authors	Objectives	Format	Approach used	Findings	Limitations	Evaluation Metrics used
(Gutub et al., 2010)	To protect the Quranic image from tampering and copyright	image	Steganography	A technique based on steganography has proposed to protect the Quranic document.	Limited to image only. There is no experimental evidence to prove this technique is prone to various tampering attacks.	There is no specific metric, taken for evaluation purposes.

Table 2.2: Continued

Authors	Objectives	Format	Approach used	Findings	Limitations	Evaluation Metrics used
(AlAhmad et al., 2013a)	To detect tampering of a document	image	Cryptography and Hashing.	The hybrid approach of AES and RSA is proposed to protect the text documents from tampering	1. Lack of proper methodology. 2. No pseudo code of the hybrid algorithm mentioned.	No experimental results showed
(Alsmadi & Zarour, 2015)	To detect and authenticate Quranic verses	text	Hashing	A method based on hashing to authenticate and verify Quranic verse.	1. Suitable for single verse only.2. The possibility of Hash collision.3. Diacritic and non-diacritic Arabic verse will give different hash values.	No performance metric evaluated. Evaluation is based on comparing hash values of different text.

The advantages and drawbacks of all the standard approaches discussed are now listed in Table 2.3.

Table 2.3: Advantages and drawbacks of standard approaches

Image-based Approaches	Advantages	Drawbacks
Watermarking	Has the additional requirement of robustness against possible attack (Xuehua, 2010)	A Lot of attacks possible in the form of geometric, noise and other related attacks
Cryptography	The main task is to ensure users able to communicate securely over an insecure channel (Coron, 2006)	More suitable for network related attacks as compared to digital documents (Delfs & Knebl, 2015)
Steganography	The message can be sent without any suspicion. Suitable for securing transmitted messages involving encoding and decoding process	Less secure. Not suitable for preserving the integrity of sensitive documents like image or text (Cole, 2003)
Digital signature	Very efficient in legally binding documents	Both senders and recipients need to buy digital certificates from trusted certification authorities

Among all these approaches, we find watermarking as the potential approach for an image-based format that can be used to develop complete protection and authentication system for Quranic materials. Thus, based on a review of related works presented in Table 2.1, research works involving watermarking for content integrity protection of the Holy Quran are explored based on important factors of imperceptibility, robustness, security and computational cost in Table 2.4.

Table 2.4: Important factors for preserving the content integrity of Digital Quran in image format

Watermarking approaches	(Tayan et al.,	(Tayan et al., 2014)	(Syifak Izhar et al., 2013)	(Kurniawan et al., 2014)	(Kurniawan et al., 2013b)	(Gutub et al., 2010)	(Laouamer & Tayan, 2013)
Imperceptibility	✓	✓	×	✓	×	×	✓
Robustness	×	×	×	✓	✓	×	✓
Security	×	×	×	×	✓	×	×
Computational cost	✓	✓	✓	×	×	×	×

From the above analysis, it can be observed all related works do have several drawbacks in terms of security, robustness and other related issues. The marks (✓) denote a particular factor has been considered and (×) depicts a particular factor not considered.

More attention should be given to improve imperceptibility, robustness, security and computational cost. All these methods need to be tested for robustness with respect to transformations using different experimentations and other related parameters mentioned above.

2.3.2 Content Integrity Authentication

Content integrity authentication refers to those approaches that can be used to determine the authenticity of DHQ content. It constitutes a key issue in the study or analysis of any Quranic online content. Based on the mode of operation, content integrity authentication can be divided into two sub-categories:

2.3.2.1 Pattern-based approaches

In order to check the authenticity of the available content, the first step is to search a specific keyword and match the pattern with the verified content. In the pattern-based approach, one of the fundamental requirements is the availability of verified content. If there is no verified content, then the data integrity check is bound to fail. We have classified the pattern-based approaches into two sub-categories:

(a) ***SQL queries***

SQL constitutes a widely-used standard database (Date & Darwen, 1997). SQL is a query-based scripting language. There are queries available in SQL which are used for searching particular patterns. Some of the powerful search operators in SQL based on B+ tree structure is the SELECT and the LIKE operator. However, the overall trend is moving towards regular expressions which use a prefix and suffix approach for searching specific patterns (Alshareef & Saddik, 2012; Date & Darwen, 1997).

*(b) **String Matching algorithms***

String/pattern matching algorithms can be divided into exact matching and approximate matching algorithms. There are many algorithms under these categories which can be used to search for a particular pattern. Boyer-Moore, Rabin Karp, and KMP algorithms are considered as standard string matching algorithms (Aho & Corasick, 1975; Navarro, 2001).

These approaches are limited in verifying the authenticity of specific and short patterns only. These approaches might not be feasible to determine the authenticity of one complete document or book due to the overhead associated with search time. For authenticating larger documents, whole-document based approaches are more feasible.

2.3.2.2 Whole-Document based approaches

In the whole document-based approaches, large documents are checked for their authenticity. We have identified the three best approaches to determine the authenticity of large documents as follows:

*(a) **Hashing Process***

In hashing, a string of characters is transformed into a specific key to represent the original string. To check the integrity of that particular string, the hash value is calculated and compared. If the hash values are identical for both documents, the data is authentic. In case the hash values do not meet, the string has been modified and is not authentic. Watermarking, Hashing or Message Digest (MDs) refers to a similar method (Techtarget, 2017). These techniques are primarily concerned with accuracy rather than performance overhead. However, hashing approaches have the problem of initially exchanging 'public-keys' between many communicating parties. Some of the most commonly used hash approaches are Message Digest 2 (MD 2) (RFC 1319), MD 4 (RFC 1320), MD 5 (1321), Secure Hash (SHA) and keyed-Hash Message Authentication Code (HMAC).

The MD 2, MD 4 and MD 5 are also known as message digests and are suitable for hashing digital signatures into shorter values. Similarly, the SHA family is suitable for allowing a larger message digest (Techtarget, 2017).

*(b) **Brute Force Process***

In the brute force approach, each character is checked one by one to be sure of data correctness and accurateness. Although this approach is quite simple and secure, it can take lots of time and space to process large files. However, this approach is efficient for short strings and produces accurate results.

*(c) **Watermarking***

Watermarking can also be used for the data integrity check. A watermark or logo is used to authenticate a particular document or image.

Summary of approaches to preserve and verify the content integrity of the Holy Quran is given below:

Having a similar objective, the study of (Nisha et al., 2014) examines different search engines and assesses their limitations. This study proposes the new search engine 'Truth-search-now'. Five search engines are benchmarked with respect to Islamic content, i.e. The Islamic search, IntoIslam, Search-truth, IslamiCity, and Allah.pk. They are evaluated based on the time taken by each search engine to conclude a particular query. However, no details are provided as to the manner in which the evaluation is conducted. Furthermore, the study fails to offer any verifiable experimental proof including the algorithm. Moreover, the proposed site (www.truth-search-now.com) online could not be located (Nisha et al., 2014).

In order to authenticate the text in the Quranic verse search, (Alshareef & Saddik, 2012) proposed a framework for Quranic verse detection. The concept is to take a Quranic verse as input and the result displays whether the verse is authentic or not. The two major components are Quranic quote filtering and verification mechanism. In Quranic quote filtering, all Arabic diacritics and special symbols are removed as they limit the traditional search engines, yet no proof or justification of this claim is given. Subsequently, the Quranic verification mechanism is applied using regular expression SQL query to verify the text. Selected verse samples are used to evaluate their authenticity. Additionally, selected Arabic words are used, and the results are compared with three Quran search engines i.e. Muslim-web, ketaballah and holy Quran. The proposed algorithm shows 89 % accuracy as compared to the other search engines. However, the accuracy is expected to vary if any algorithm is used on a larger dataset. Specifically, this algorithm accuracy is expected to decrease if applied to a large Arabic data set due to the fact that regular engines use a prefix-suffix approach.

In order to verify Arabic verses along with diacritics and symbols, the study of (Alginahi et al., 2013) has proposed an algorithm able to detect complete and partial verses. For evaluation, the two parameters 'verified and authenticated' and 'tampered with' are considered. Nevertheless, no details are given on the algorithm mechanism. Also, the flowchart indicates a simple SQL approach is used to select the query which is a rather inefficient approach that needs to specify a particular location first. Besides, the proposed algorithm considers all symbols and diacritics in the flowchart, and the algorithm removes all symbols and diacritics before conversion to Unicode format (Alginahi et al., 2013).

A new framework is introduced to detect and authenticate Quranic verses in a text extracted from online sources such as forum posts (Sabbah & Selamat, 2013). The objective is to increase the detection accuracy of the diacritic text. The achieved accuracy on average is 62%, while precision and recall are measured at 75 % and 78 %, respectively. The proposed framework involves an extractor which extracts three lists from the Quranic script, i.e. Distinctive Quranic diacritical words, distinctive letters, diacritics, and symbols. Each letter, diacritic, and symbol are given a distinctive weight. After assigning weights to all words, they are grouped into a character and a diacritic set. However, this algorithm does not work on the non-diacritical text, and there is too much overhead associated while calculating weights and dividing the respective verses into two groups. The complexity of the algorithm increases if the texts contain many diacritical signs.

The summary of the work done to preserve and verify the binary (text) format of the DHQ is shown in Table 2.5.

Table 2.5: Advantages and drawbacks of String/pattern matching approaches used in authenticating/protecting DHQ content

Authors	Objectives	Approach used	Findings	Limitations	Evaluation Metrics used
(Alshareef & Saddik, 2012)	To detect fake Quranic verses	A regular expression approach using SQL queries.	A framework is proposed which can authenticate Quranic verses.	For finding a particular verse, the user needs to enter surah-name also which is one of the major limitations of this approach.	For evaluation purposes, the accuracy metric has been taken and benchmarked with Ketaballah.net, Muslim-web.com, Holyquran.net.
(Sabbah & Selamat, 2013)	To detect and authenticate Quranic verses	SQL Query approach	A framework has been proposed to detect and authenticate the Quranic text.	1. Not suitable for non-diacritical text.2. The complexity of the algorithm will increase with more complex diacritical verse.	Accuracy, precision, and recall have been taken as evaluation parameters
(Alginahi et al., 2013)	Verification of quoted Quranic text with diacritics and Tajweed symbols	SQL Query Approach	The algorithm has been proposed for the detection of particular Arabic verse.	Not efficient as user needs to know the particular verse to be verified is from which chapter. The algorithm will fail if there is the last verse of one Surah and the first verse of consecutive Surah.	Two parameters, i.e., “Verified and authenticated” and “tampered with” are taken.
(Nisha et al., 2014)	To study different Quranic engines and their	ASP. Net platform has been used to	A new search engine “truth-search-	Could not locate the proposed engine online.	Performance of 5 search engines with respect to Islamic content has been evaluated by

Table 2.5: Continued

Authors	Objectives	Approach used	Findings	Limitations	Evaluation Metrics used
	limitations with their data mining ability	develop the system	now" has been proposed.		measuring search time.

The main drawbacks of the standard approaches used to determine the authenticity of DHQ above in section 2.3.2 are now listed in Table 2.6.

Table 2.6: Drawbacks of content integrity authenticity approaches

Approaches	Drawbacks	Recommendation
SQL query	Linear time complexity if Index not provided. In case Index is provided, the user needs to make sure the input verse belongs to which chapter. Prone to SQL injection attacks	Not Suitable for Searching random Quranic verses
String/Pattern Matching	Results in slower performance with respect to the processing time for Non-ASCII based texts due to use of different encoding techniques	A potential approach to search and verify Quranic verses.
Hashing	Might result in a hash collision. Overhead associated in case many single Quranic verses need authentication.	Not recommended for authenticating single Quranic quotes. Recommended for whole/multiple pages.
Brute force	Linear time complexity (more processing time).	An efficient approach for verifying Quranic verses provided time complexity can be reduced.
Watermarking	Prone to many attacks like geometric, noise if image logo embedded in the plain text. In the case of bits embedded in the plain text, the overhead associated with making it secure.	Not recommended based on a number of possible attacks and overhead associated with watermarking for plain text.

Based on the analysis of all standard approaches that have been used for determining the authenticity of the text-based Quranic verses, we have identified SQL and pattern based as promising approaches to achieve the task of authenticating single and multiple Quranic verses. Besides, the review of related works presented in Table 2.5 indicated that the main focus of SQL based approaches has been on accuracy rather than time complexity.

Table 2.7: Important parameters to determine authenticity of Digital Quran in binary format

SQL based approaches	(Alginahi et al., 2013)	(Sabbah & Selamat, 2013)	(Alshareef & Saddik, 2012)
Accuracy	×	✓	✓
Time complexity	×	×	×

Table 2.7 shows that existing works using SQL approaches have several drawbacks for authenticating Quranic content. Firstly, one of the important performance measures i.e. time complexity has not been taken into consideration. Secondly, two popular algorithms i.e. Linear search and Binary search (Welling & Thomson, 2003) have been used for searching and verifying the verse from a given database. Linear search is also known as a sequential search algorithm. It works by comparing each element of the array one by one in a sequence until a match is found. Thus, the time complexity of the linear search algorithm is O(n) time (where n is the number of elements in the array). On the other hand, Binary search reduces search time to half provided the list is sorted. It works by comparing the given element with a middle element in the list. If both the elements match, the search process completes. In case, no match is found, it checks whether the given element is smaller or larger than the middle element. If the given element is smaller than the middle element, the search is confined to the left sublist and the same procedure is followed as mentioned above till a match is found. In case, the search element is larger,

then the right sublist is searched until the complete match is found. Thus, it can be observed Linear search algorithm increases search time while binary search algorithm needs an index approach with efficient queries which limits the capability of searching to specific index only Hence, an alternative approach is needed to enhance the search and verification phase for searching and authenticating Quranic texts.

2.4 Open issues, challenges, and possible solutions

a. *Need for an appropriate authentication approach:*

The previous section has shown different approaches that can be used to preserve and authenticate digital Holy Quran on the internet. Watermarking, SQL, and String matching techniques seem to be promising approaches that can be used to develop the complete Quran authentication system. However, keeping the sensitivity of the Holy Quran into consideration, developing of such an approach that can perform better in terms of security, robustness, computational costs, imperceptibility and other related parameters compared to existing methods, is still an open issue and need more research efforts. In this case, watermarking, string matching and SQL approaches can be explored further for more efficient solutions.

b. *Availability of Quran and Hadith Apps in Mobile Platforms:*

The number of mobile users is continuously increasing as shown in Figure 2.5 based on data made available (gs.statcounter, 2018). In fact, the number of mobile users has already crossed the number of desktop users since 2014. The popularity and use of mobile platforms like Android, IOS, and Symbian are rapidly increasing.

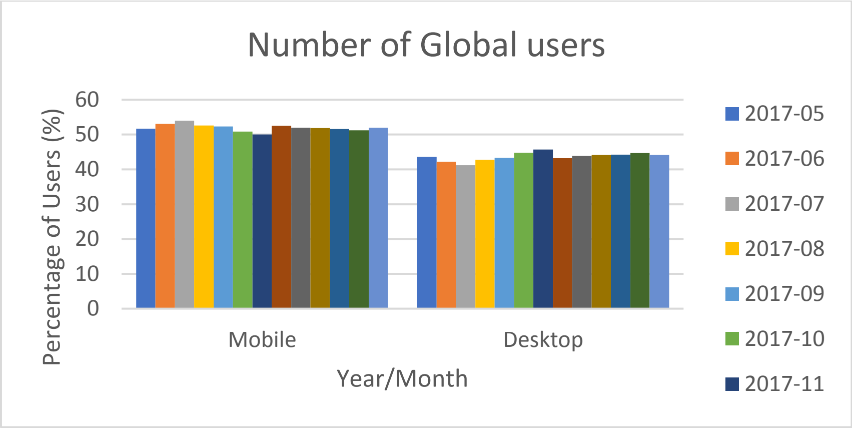


Figure 2.5: Number of mobile users (http://gs.statcounter.com, 2018)

The easy access to the Quran and *Hadith* applications on those platforms is one of the most challenging issues. Most Muslims around the world download and follow those applications blindly. There is no appropriate mechanism which can verify the reliability of these applications. This trend is worrisome and calls for proper measures to be used. One possible solution is to identify all Quranic applications. Once identified, each application can be evaluated manually by Islamic scholars. However, this approach is admittedly very tedious and cumbersome. Another approach is to develop a system, which can access all the content of an application automatically and verify it. Additionally, signature verification may be followed by a collaboration with a Google team and Islamic scholars to help in verifying authentic applications. This issue constitutes indeed a major challenge and calls for more rigorous analysis and research.

c. *A reliable database of authentic and verified Quran and hadith content:*

Hadiths are sayings and reported actions of Prophet Mohammad (PBUH) and constitute the second most valuable reference after the Holy Quran. When carrying out research and checking of the validity of different approaches, knowledge of a reliable and authentic database is of utmost importance. Although there are reliable databases available such as tanzil.net, one can never be sure whether or not each piece of information displayed on the website is reliable. A possible solution to this problem is to develop an authentic database and have the content verified by an authorized Islamic religious body. The first challenge, however, would be to make all *Hadith* collections available when creating such a database.

d. *Identification and Blacklisting of Fake Islamic Websites:*

Many websites claim to be managed by Islamic scholars or other Islamic religious bodies. However, in actual practice, these websites have been created for the sole purpose of misguiding Muslims and non-Muslims alike by offering false information through fabricated *Hadiths* and misinterpreted Quranic verses. This issue is again a very challenging problem. These websites are developed by unknown individuals or parties who quote Quranic verses out of context and mislead those users who are still unfamiliar with the Quranic message. The following website is one example of such abuse (Religion of peace, 2018). On this website, all Quranic quotes have been quoted out of context and have been developed to give the impression that Islam promotes violence. There are hundreds of such websites. One way of addressing this grave problem is to blacklist these kinds of websites after proper identification using suitable keywords or by developing an automatic system based on a key-word search which can help in identifying those websites so that appropriate action can be taken.

e. *Fabricated Hadith Detection System:*

There are many *Hadith* collections containing thousands of traditions made available online which makes it nearly impossible for the average Muslim to distinguish between authentic and fabricated ones. Thus, many fabricated *Hadiths* are readily accessible online with no proper way to immediately verify the authenticity of the same. These fabricated *Hadiths* can be found on social media websites, online blogs and on regular websites. Again, this problem is too complex and challenging as it is not possible to stop people from posting fabricated *Hadiths*. A possible solution would be to develop an authentic database of all *Sahih* collections certified by an established religious body. This database can then be used to develop an authentic *Hadith* based website where users can check whether a particular *Hadith* they have come across on the internet is authentic or fake. Since the Muslim community is a global community, such a system would have to be multilingual.

f. *Open Source Library:*

There is a need for an online library where all proposed algorithms with respect to preserving data integrity of the Quran and *Hadith* can be made readily available to the researchers. Such a reliable online library can be of great assistance to future researchers who wish to test those algorithms and propose more efficient approaches with high accuracy and precision. GitHub is one of the most useful and efficient databases in this respect.

g. *The message of Peace System:*

Islam is the religion of peace. Given the negative and misleading associations of Islam with violence, terrorism, and aggression, which dominate most public media today, the original message of Islam as espoused in the Holy Quran must be made available to all people around the world. Although there are already numerous social media websites available which are trying to accomplish just that, their scope remains limited. A separate system needs to be put into place such as online teaching sites, where all basics of the Quran along with important moral and ethical guidelines can be taught to non-Muslims and Muslims alike.

h. *Arabic Text Pattern Recognition Using Images:*

Pattern recognition constitutes a highly researched area of study. The process of verifying content in the form of images, particularly in Arabic, remains a challenging and open issue. There are numerous minor problems in this area, such as the extraction of overlapping Arabic characters in an image and text retrieval from images using different writing styles. This area needs further research in order to identify further issues. Possible solutions may lie in the use of segmentation techniques and machine learning approaches (Zerdoumi et al., 2017).

i. *Numerical Structure of the Holy Quran:*

There have been some observations that the Holy Quran is numerically structured based on the number 19. The opening verse of Holy Quran, i.e." In the Name of God, Most Gracious, Most Merciful" (بسم الله الرحمالرحيم)is of 19 Arabic letters. Similarly, the first chapter of the Holy Quran revealed to Prophet Mohammad (PBUH) i.e. Chapter 96 (Embryo) also contains 19 verses. There are many such examples in Holy Quran related to number 19. Hence, it will be interesting research work to authenticate the Holy Quran based on numerical analysis.

j. *Expert Real-time Quranic Verse Detection System:*

The ready availability of Quranic content on the Internet has made it very convenient for Muslims and non-Muslims to read up on any issue online. However, this convenience comes at a cost. Since the original Quran was revealed in Arabic and the majority of Muslims are not Arabs, and often only know how to recite selected verses from memory, they are dependent on translations. The reader unfamiliar with the Arabic language will need the help of diacritic signs and symbols to read the script correctly. Thus, there is the need to develop an efficient system that can detect and inform users of possible changes in a specific verse. In recent years, some efforts have been made in respect to verse verification, but they are not efficient enough in terms of accuracy and precision. A possible solution is to identify suitable encoding techniques and identify weaknesses in the present string matching approaches.

k. *Authentication of Quranic texts available in different styles:*

Quranic texts can be found written in many different styles like Uthmanic, Warsh, plain texts without diacritics, plain texts with some diacritics and so on. Hence, it is quite interesting to authenticate the most popular and standard styles of the Quran using one common database. It will avoid the need of having more than one system to authenticate different styles of Quranic texts. Initially Uthmanic and plain Quranic texts can be taken into consideration due to their popular usage throughout the world (Hakak et al., 2018a).

The aforementioned points constitute a selection of the major challenges and open issues together with possible solutions and recommendations for future research. It is ceded that there are many other issues which can be addressed by putting more efforts in analyzing research done in the area of authenticating religious texts like DHQ.

2.5 Conclusion

This chapter reviews recent studies on one of the most sensitive diacritical texts i.e. Digital Holy Quran integrity protection and authenticity. There are numerous issues in this area which call for a resolute and timely response in the form of intensified research efforts. Quran authentication and protection faces many challenges, foremost improving the accuracy and precision of text detection. In this chapter, the most common challenges are pointed out and solutions are proposed. A brief overview of the existing research in this field is given, the possible limitations are assessed, and their findings evaluated. The promising directions which future research should take as discussed in section 2.4 include the call for a reliable universal database of authentic and verified Digital Quran and

Hadith content, another major task for future researchers is to develop the Expert Real Time Quranic Verse Detection System with improved time complexity and accuracy.

Hence, based on the literature review, the problem of authenticating different styles of Quran along with efficient time complexity needs immediate attention. The same has been addressed in this research.

CHAPTER 3: RESEARCH METHODOLOGY AND FRAMEWORK

3.1 Introduction

This chapter briefly describes the general methodology to complete the proposed research. A framework focusing on both authentication and protection of sensitive diacritical content is proposed. The case study of DHQ has been taken for evaluation purposes. The aforementioned framework consists of two main phases which are authentication and protection. The research challenges of authenticating the two most widely used diacritical Quranic script styles by using a common database and achieving optimal retrieval of diacritical texts are presented in Section 3.2. The proposed framework is briefly explained in Section 3.3. Since the focus of this research is on the authentication phase, the aim of showing the complete framework is to describe the working of a full expert system that can authenticate and protect SDC such as Quranic text. The proposed methods, experimental setup, and evaluation metrics are briefly discussed in Section 3.4.

3.2 Research challenges

Based on the literature review, it is noted that the existing systems suffer from several serious challenges as shown in Figure 3.1. The problem of authenticating more than one script style and improving search time is further divided into the preprocessing phase and the authentication phase (comprising of retrieval and search process). A description of those challenges is given in Figure 3.1.

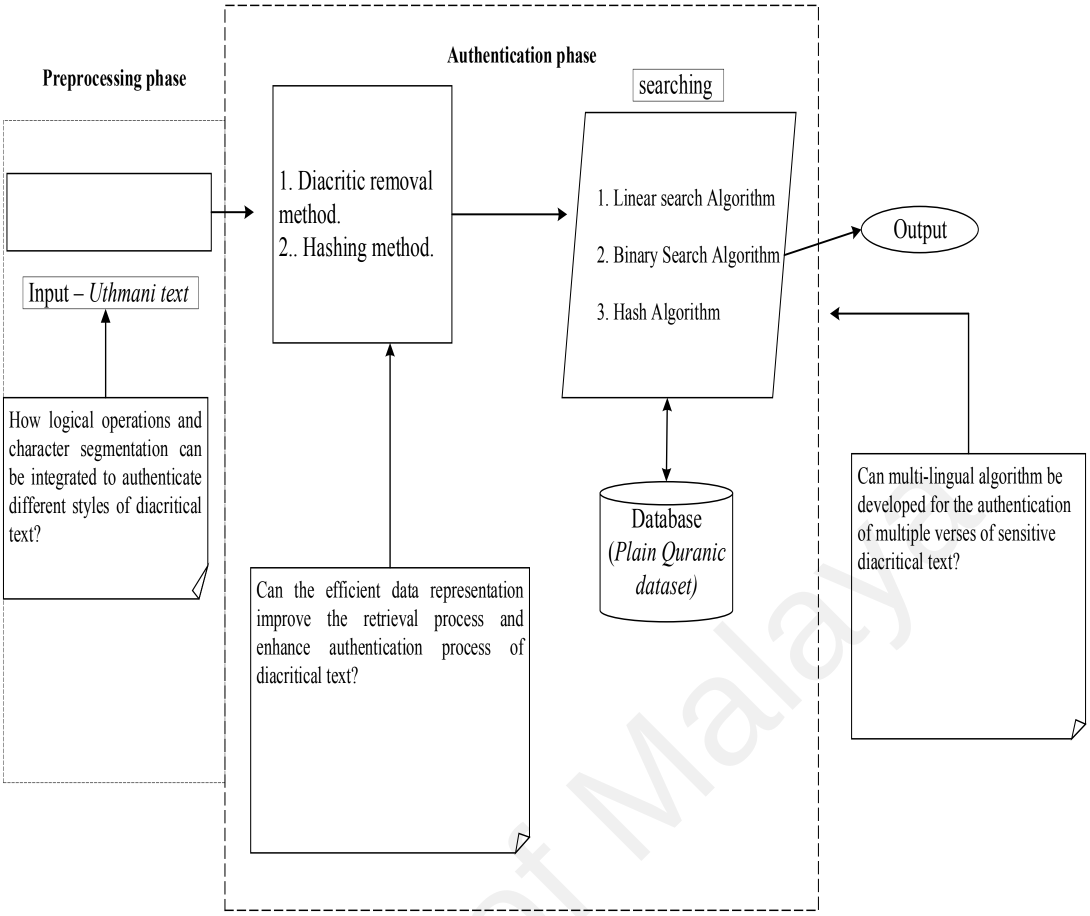

Preprocessing phase
Authentication phase
searching
1. Diacritic removal method.
2.. Hashing method.
1. Linear search Algorithm
2. Binary Search Algorithm
3. Hash Algorithm
Output
Input – Uthmani text
How logical operations and character segmentation can be integrated to authenticate different styles of diacritical text?
Can the efficient data representation improve the retrieval process and enhance authentication process of diacritical text?
Database (Plain Quranic dataset)
Can multi-lingual algorithm be developed for the authentication of multiple verses of sensitive diacritical text?

Figure 3.1: Authenticity challenges

a. Preprocessing phase: The first and the foremost challenge with respect to the preprocessing phase is the type of input (Uthmani or Plain script style) and the character segmentation. As shown in Figure 3.1, all the existing approaches reviewed in Chapter 2 fail to authenticate the diacritical verse if the reference database script style differs from the input style. If the input is in the Uthmani script as shown in Figure 3.1 and the database is in plain script, the existing systems generate the wrong output and cannot identify the correct verse.

b. Authentication phase: In most of the existing studies, the diacritics in the pre-processing stage are removed to guarantee the efficient retrieval of diacritical Quranic texts for authentication purposes. Even most popular search engines fail to retrieve diacritical verses. This observation raises several questions, for example, whether diacritics affect the retrieval and search process. Also, worth investigating is whether the efficient representation of diacritical Quran texts improves the retrieval and search process thereby improving the authentication process.

As already discussed in Chapter 2, the two most popular algorithms (linear and binary search) are used to authenticate Quranic verses. In some cases, hash algorithms are also used. Among the main limitations of these approaches is increased time complexity. In the binary search algorithm, the chapter number of a particular verse needs to be included in order to be searched and verified. In order to search and verify a verse *V*, the user needs to know the exact chapter from which that verse belongs. Similarly, hash algorithms can result in hash collisions, and a linear search algorithm is more suitable to authenticate single Quranic verses rather than multiple interconnected verses with increased time complexity. A detailed discussion of authenticating single and multiple interconnected verses is given in Chapter 5 and 6 respectively.

Based on the above-mentioned challenges, a framework is proposed that can help address the above issues. The proposed framework is explained in the next section.

3.3 Proposed authentication framework

The proposed framework (Hakak et al., 2018b) is based on binary data meaning text as shown in Figure 3.2. This framework can be used to identify incorrect Quranic diacritical verses from public online sources.

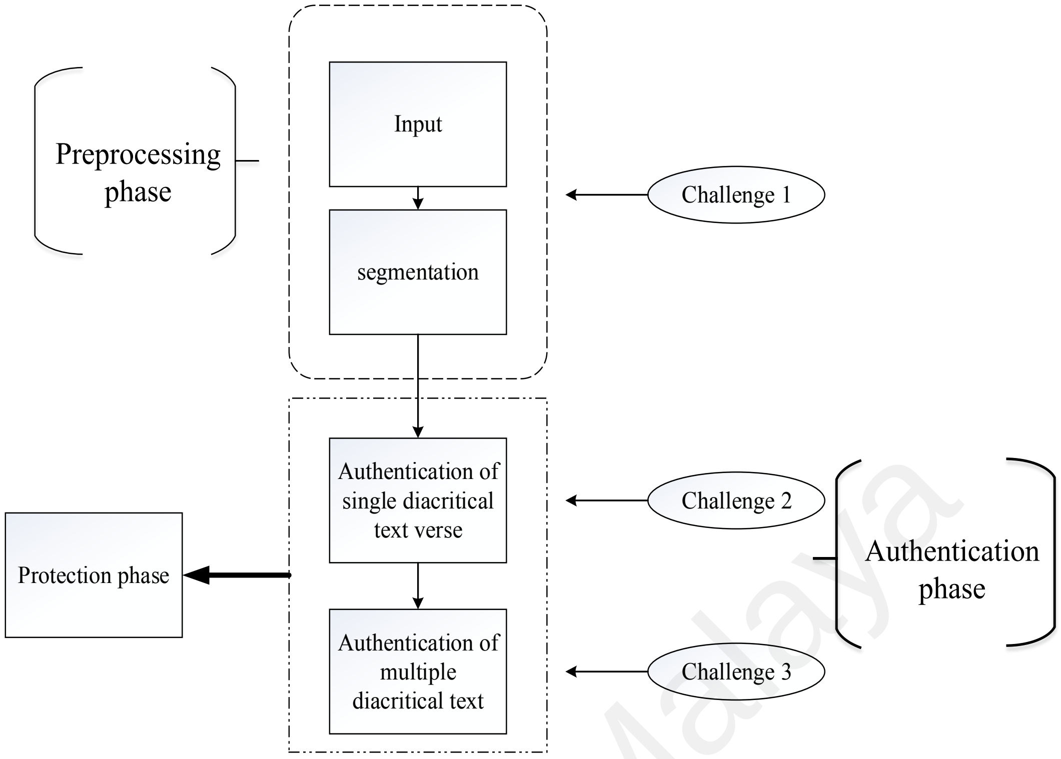


Figure 3.2: Proposed Quran authentication and protection framework

The proposed framework consists of the content integrity protection phase and the content integrity authenticity phase. The purpose of the protection phase is to protect verified content from subsequent tampering and make authenticated content available online.

Since the focus of most current research in this field is to determine the authenticity of diacritical texts such as DHQ, only the steps involved in the authentication phase are presented. The content integrity authenticity phase comprises the pre-processing phase and the authentication phase.

Preprocessing Phase: In this component, the user enters diacritical Quranic verse to be authenticated. For the verses written in the Uthmani style or plain Quranic style, the residual approach converts the input into standard form (as in the same form of database). It must be noted here, the standard form of Quranic dataset involving 34 chapters of the

Holy Quran was created from Tanzil.net (tanzil.net, 2016) and verified by the experts from the faculty of Islamic studies, University of Malaya. The created dataset is still under progress till remaining chapters of DHQ are also included. The proposed approach is discussed and evaluated in Chapter 4 along with related works and future research directions. This approach addresses the challenge of authenticating Uthmani and plain script using a common database.

Authentication phase: To authenticate single DHQ verses, a new method of data representation is proposed and evaluated that is specifically designed for sensitive diacritical texts like the Digital Quran with the aim to improve the retrieval efficiency and improve search time. The proposed representation can retrieve and authenticate single Quranic verses accurately and without removing the diacritics first. The need of efficient data representation is concluded on the experiments carried out using factorial design, a mathematical model that helps to determine the factors influencing the output. The complete methodology of the proposed representation and factorial design is presented in Chapter 5 along with other related studies.

To addresses the limitation of the single verse authentication (challenge 2) of the conducted research, which is to reduce the search time and authenticate more than one interconnected diacritical Quranic verse, the pattern matching based approach is proposed. The complete methodology and benchmarks are discussed in Chapter 6.

3.4 Proposed Methods and Experimental setup

The overall methodology of the conducted research along with the different methods proposed for the phases shown in Figure 3.2 (pre-processing & authentication) is depicted in Figure 3.3. For the preprocessing phase, a residual method is proposed and for the authentication phase, the data representation and pattern-based approaches are proposed. The flow of the conducted research (as shown in Figure 3.3) is as follows:

The required input content (in Uthmani or plain script style) is converted into a common format using a residual approach. This is achieved by identifying the differences between the two styles and substituting the different symbols like complex *alif* and other related symbols that do not alter the meaning. Given that no existing study has yet addressed this issue, the proposed approach is justified by the documented failure of the following DHQ authentication approaches: Quran quote verification (QQV) algorithm (Alshareef & Saddik, 2012), Quran verse verification and authentication algorithm (Alginahi et al., 2013) and the hashing approach (Alsmadi & Zarour, 2015). These existing approaches are not able to authenticate more than one style of Quranic script content. A further description is given in Chapter 4.

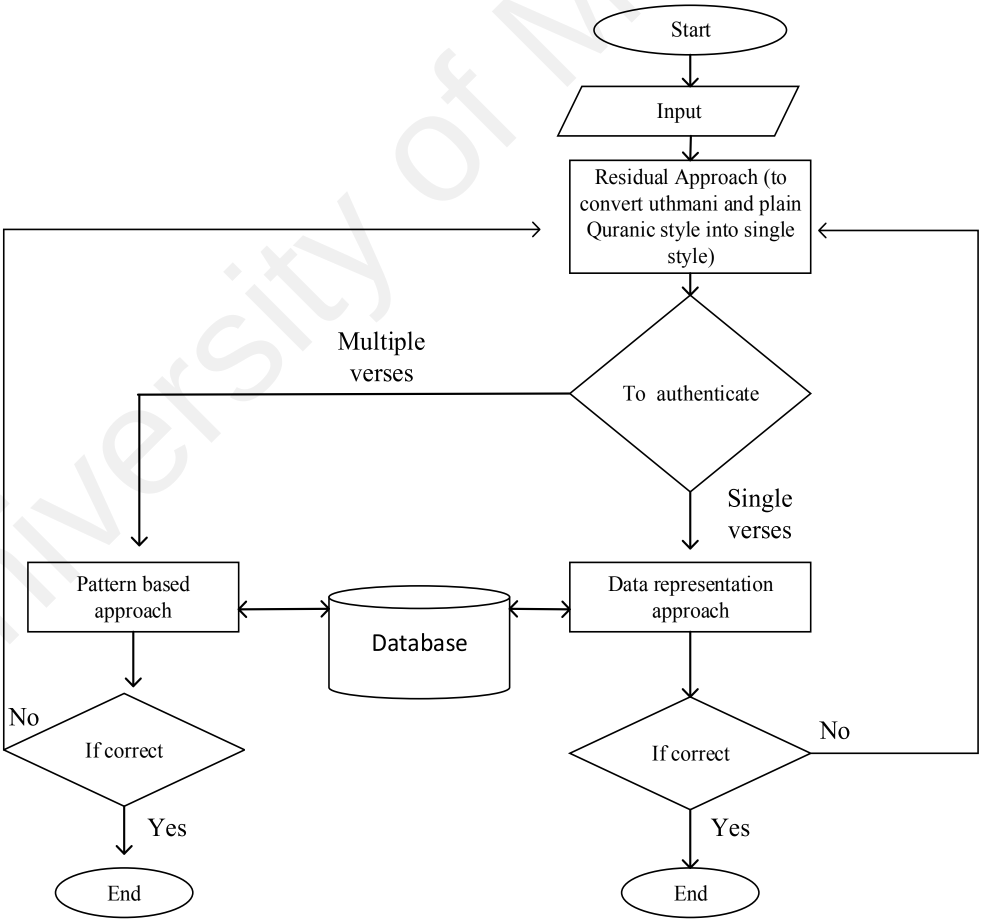


Figure 3.3: Overall methodology of the conducted study

Once the input is converted into a standard script style (format), the second phase or the authentication phase commences in order to improve the retrieval and search process. For this purpose, a new data representation is proposed that arranges the Quranic verses in a database based on the first character. This representation improves the retrieval and search process to the maximum extent. The proposed approach is then compared to the Quran Quote Verification (QVT) algorithm (Alshareef & Saddik, 2012), traditional MySQL using the linear search algorithm, binary search, and the two popular Quranic search engines Muslim Web and Search Truth (Greenspan & Bulger, 2001). The proposed data representation is limited to authenticating single verses. To overcome this particular limitation, a pattern matching approach is proposed that can authenticate more than one verse at a time. The approach is then compared to the Boyer-Moore (BM) algorithm (Boyer & Moore, 1977), the Turbo Boyer- Moore (TBM) (Crochemore, 1994), the Simple String Matching (SSM) algorithm (Al-Ssulami, 2014) and the brute force approach. The full description of the new data representation and pattern matching approach is given in Chapters 5 and 6 respectively.

For the experimental part of this study, the datasets are taken from (tanzil.net, 2016). The evaluation parameters taken include accuracy and time complexity (the time taken to retrieve and search a verse). The accuracy is calculated by dividing the number of verses found to the total number of verses present in a given dataset. The hardware requirements include an I-5 processor operating on Windows 10 and Net Beans 8 IDE.

3.5 Summary

This chapter discusses the major challenges that need to be overcome for the successful conclusion of this study and outlines the tasks and experiments required to solve them. The major challenges were modularized into two phases: pre-processing and authentication (comprising of retrieval and search process). The limitations discovered in each phase are briefly summarized in the form of a block diagram (as shown in Figure 3.1). Based on the challenges, a complete framework is proposed involving the protection and authentication phase. The solution for each respective challenge is proposed and briefly summarized along with the proposed framework (as shown in Figure 3.2). Detailed solutions are given for the preprocessing phase, and the authentication phase in Chapters 4, 5 and 6.

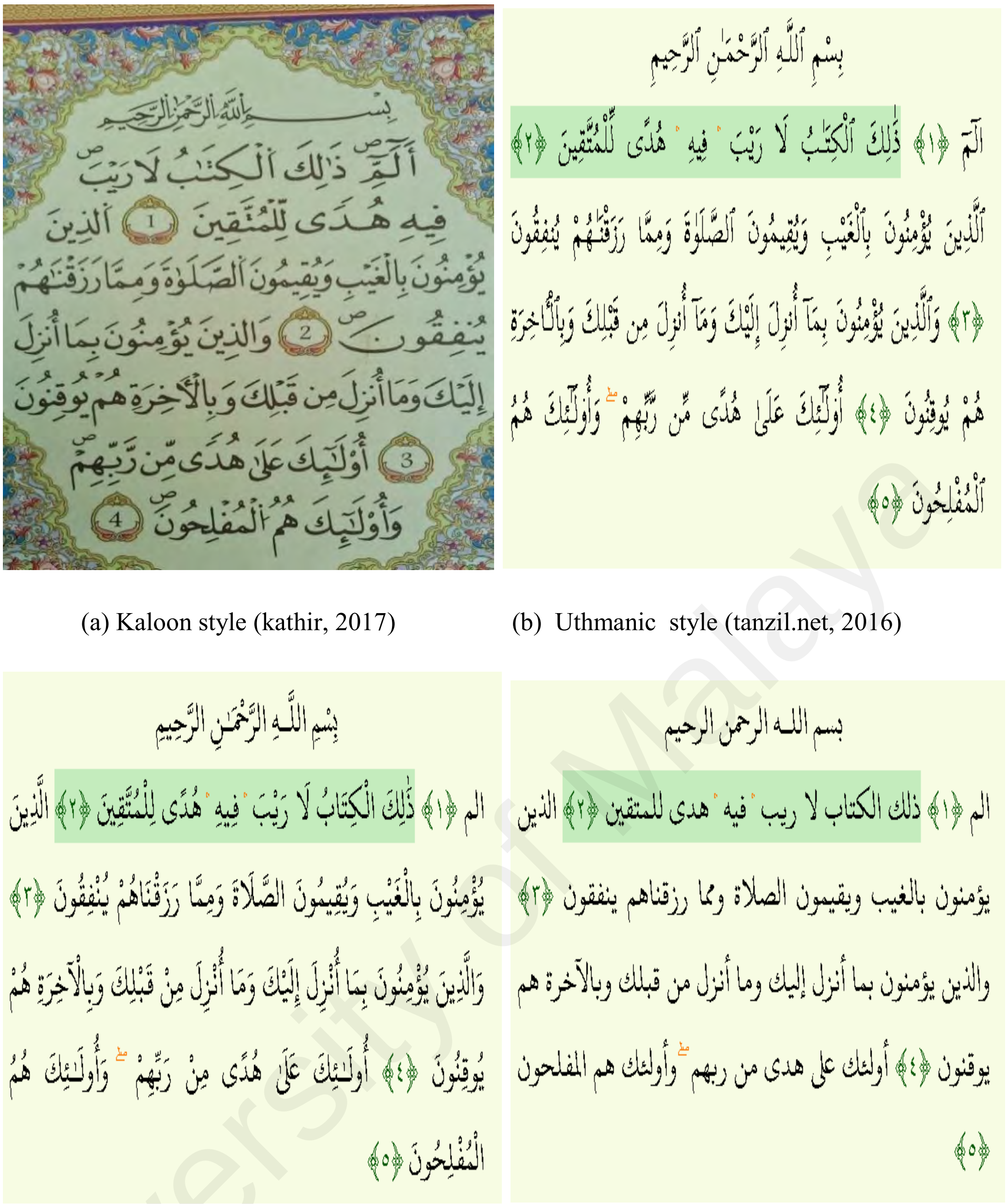
CHAPTER 4: RESIDUAL BASED APPROACH FOR AUTHENTICATING PATTERN OF MULTI-STYLE DIACRITICAL ARABIC TEXTS

4.1 Introduction

This chapter elaborates on the discussion presented in the previous chapter. The input consists verses either in Uthmani or plain script. After the required pre-processing, the text is converted into a script format similar to the format available in the database. The aim of this phase is to convert the Uthmani or plain script text into one format to render the authentication process more efficient. The chapter discusses the related work in Section 4.2. The proposed approach is explained in Section 4.3, and the experimental results are summarized and evaluated in Section 4.4. A chapter summary is presented in Section 4.5.

4.2 Related work

Digital versions of the Quran are made available online in different styles for reading purposes. As briefly mentioned in Chapter 2, the issue of credibility and authenticity is drawing more and more public attention (Alsmadi & Zarour, 2015; Hakak et al., 2017a). Since the Quran is a sensitive script, its authentication and integrity are of greatest concern (Alsmadi & Zarour, 2015; Elayeb & Bounhas, 2016; Rafe & Nozari, 2014; Sabbah & Selamat, 2015). The Quran is written in Arabic language and in different styles such as plain text (mostly used in countries like India, Pakistan and Bangladesh), Uthmanic, Kufi, Kaloon and other such styles. Several such styles are shown in Figure 4.1.


بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ
الٓمٓ ذَٰلِكَ الْكِتَٰبُ لَا رَيْبَ
فِيهِ هُدًى لِّلْمُتَّقِينَ 1 الَّذِينَ
يُؤْمِنُونَ بِالْغَيْبِ وَيُقِيمُونَ الصَّلَوٰةَ وَمِمَّا رَزَقْنَٰهُمْ
يُنفِقُونَ 2 وَالَّذِينَ يُؤْمِنُونَ بِمَا أُنزِلَ
إِلَيْكَ وَمَا أُنزِلَ مِن قَبْلِكَ وَبِالْآخِرَةِ هُمْ يُوقِنُونَ
3 أُولَٰٓئِكَ عَلَىٰ هُدًى مِّن رَّبِّهِمْ
وَأُولَٰٓئِكَ هُمُ الْمُفْلِحُونَ 4

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ
الٓمٓ ﴿١﴾ ذَٰلِكَ ٱلْكِتَٰبُ لَا رَيْبَ ۛ فِيهِ ۛ هُدًى لِّلْمُتَّقِينَ ﴿٢﴾
ٱلَّذِينَ يُؤْمِنُونَ بِٱلْغَيْبِ وَيُقِيمُونَ ٱلصَّلَوٰةَ وَمِمَّا رَزَقْنَٰهُمْ يُنفِقُونَ
﴿٣﴾ وَٱلَّذِينَ يُؤْمِنُونَ بِمَآ أُنزِلَ إِلَيْكَ وَمَآ أُنزِلَ مِن قَبْلِكَ وَبِٱلْءَاخِرَةِ
هُمْ يُوقِنُونَ ﴿٤﴾ أُو۟لَٰٓئِكَ عَلَىٰ هُدًى مِّن رَّبِّهِمْ ۖ وَأُو۟لَٰٓئِكَ هُمُ
ٱلْمُفْلِحُونَ ﴿٥﴾

(a) Kaloon style (kathir, 2017)   (b) Uthmanic style (tanzil.net, 2016)

بِسْمِ اللَّـهِ الرَّحْمَـٰنِ الرَّحِيمِ
الم ﴿١﴾ ذَٰلِكَ الْكِتَابُ لَا رَيْبَ ۛ فِيهِ ۛ هُدًى لِّلْمُتَّقِينَ ﴿٢﴾ الَّذِينَ
يُؤْمِنُونَ بِالْغَيْبِ وَيُقِيمُونَ الصَّلَاةَ وَمِمَّا رَزَقْنَاهُمْ يُنفِقُونَ ﴿٣﴾
وَالَّذِينَ يُؤْمِنُونَ بِمَا أُنزِلَ إِلَيْكَ وَمَا أُنزِلَ مِن قَبْلِكَ وَبِالْآخِرَةِ هُمْ
يُوقِنُونَ ﴿٤﴾ أُولَـٰئِكَ عَلَىٰ هُدًى مِّن رَّبِّهِمْ ۖ وَأُولَـٰئِكَ هُمُ
الْمُفْلِحُونَ ﴿٥﴾

بسم الله الرحمن الرحيم
الم ﴿١﴾ ذلك الكتاب لا ريب ۛ فيه ۛ هدى للمتقين ﴿٢﴾ الذين
يؤمنون بالغيب ويقيمون الصلاة ومما رزقناهم ينفقون ﴿٣﴾
والذين يؤمنون بما أنزل إليك وما أنزل من قبلك وبالآخرة هم
يوقنون ﴿٤﴾ أولئك على هدى من ربهم ۖ وأولئك هم المفلحون
﴿٥﴾

(c) Plain Arabic style   (d) Clean Arabic style(tanzil.net, 2016)

Figure 4.1: Different writing styles of Digital Holy Quran

As shown in Figure 4.1, all the styles presented differ in the way diacritics and other written properties, like dots, are arranged. Most of the native speakers of Arabic do not need diacritics to read Holy Quran, as shown in Figure 4.1 (d) (Farghaly & Shaalan, 2009). However, it is critical for non-native speakers to use these diacritics to recite and understand it properly (Arslan, 2015; Mohammed et al., 2015). For example, the basic diacritics of the Quran are shown in Table 4.1. If the diacritics are misplaced in a verse, the whole meaning of the verse is altered (Arslan, 2015; El-Defrawy et al., 2016; Hakak et al., 2017c). However, most of the existing approaches related to the authentication of DHQ texts remove such diacritics to improve retrieval results (Ismail et al., 2014; Kanan & Fox, 2016; Khalaf et al., 2014). A list of the diacritic symbols and Tajweed symbols (the set of rules related to the recitation) indicate where to stop recitation and are shown in Tables 4.1 and 4.2, respectively.

Table 4.1: Main Arabic Diacritics (Alshareef & Saddik, 2012)

Futtha	◌َ	Tenween Futtha	◌ً
Thummah	◌ُ	Tenween Thummah	◌ٌ
Kusrah	◌ِ	Tenween Kusrah	◌ٍ

Table 4.2: Tajweed Symbols

Continuing is better	◌ۖ
Must stop	◌ۜ
Topping is better	◌ۗ
Must continue	◌ۙ

Alshareef et al. (Alshareef & Saddik, 2012) have proposed the Quran Quote verification algorithm which removes all diacritics from the input verse and authenticates the verse using a diacritic free dataset. Similarly, Alginahi et al. (Alginahi et al., 2013) have proposed an algorithm for verifying Quranic verses online. The approach ignores diacritics and tashkeel (vowel marks) for efficient verification. It converts bits of text to UTF format and authenticates through a UTF database. Alsmadi et al. (Alsmadi & Zarour, 2015) used a hashing approach for authenticating Quranic verses without removing the diacritics. Most of the previous studies have focused on authenticating one single writing style. However, all these approaches are prone to fail as soon as they have to deal with different styles. Such approaches only work when the input verse and the database contain the same style. Adding or deleting one single symbol results with either a different meaning of the entire verse or causes authentication issues. One example to illustrate the difference between Uthmanic and plain text is shown in Figure 4.2, where the differences are marked by ovals.

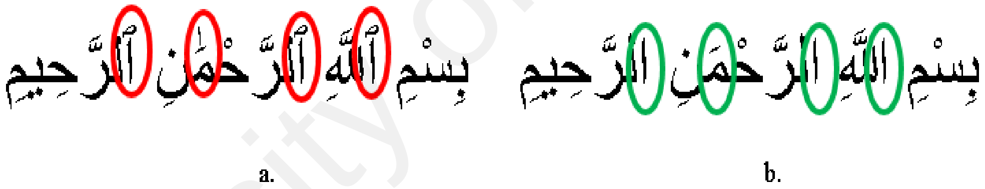


Figure 4.2: (a) Uthmanic style (b) Plain writing style

It is observed from the Uthmanic style, that the letter *alif* (أ) encircled with a circle is written differently compared to the plain style equivalent. It is also noted that '*alif*' written in plain style is simpler in form (ا). In general, in case of the Uthmanic verses, a small *alif* (ٰ) appears over the letter *mim (م)* to express the sounds of the letter, whereas plain script does not include those symbols. However, both verses written in the Uthmanic and plain script are correct. Since there are no existing algorithms that can authenticate different Quranic writing styles from a common reference/benchmark (dataset created and verified manually), a new approach is required that can authenticate different styles using one common database.

From the above discussion, it is evident that there is no effective approach to authenticate different styles of Quranic texts using one common database. Hence, the focus of this input phase is to propose an approach that can solve the authentication issues of Quranic texts available online which are written in different styles. In this phase, Uthmanic and plain Quranic verses are converted into one common style using the residual approach as explained in the subsequent sections.

4.3 Proposed approach

As discussed in Chapter 2, we consider authenticating Uthmanic and plain Quran text writing styles, as both are widely used for communication through the web or email (Hakak et al., 2018a; tanzil.net, 2016) using residual approach. A residual approach can be defined as an approach in which output is obtained, from the bitwise difference of two inputs. Based on an output, one can determine the difference between two inputs. For each verse in the Uthmanic style, the proposed method determines the residual by performing logical operations at bit level with the reference database. Using the logical XOR operation, the difference between the two styles (i.e. residual) is analysed in a bitwise manner. XOR is a logical digital gate with two or more inputs with one output. When the output is 1 (one), it implies there is a difference in input. In case, the output is zero (0), the inputs are the same (Mano, 2017). Hence, using the residual approach, Quranic verses with the output of 1 were analysed and differed letters were substituted with a simpler version of the letter. To be more specific, the verses resulting in an output of 1 indicated that, the input verse is different from the reference database. Hence, those input verses were analysed manually to find out the difference from the reference database and the appropriate substitution procedure was carried out to make input consistent with the reference database. The existing string matching algorithm was applied to authenticate the converted Uthmanic/plain Quranic verse. Using string matching, a given pattern is searched character by character from a given text (Boyer & Moore, 1977). The flow of the proposed method is shown in Figure 4.3.

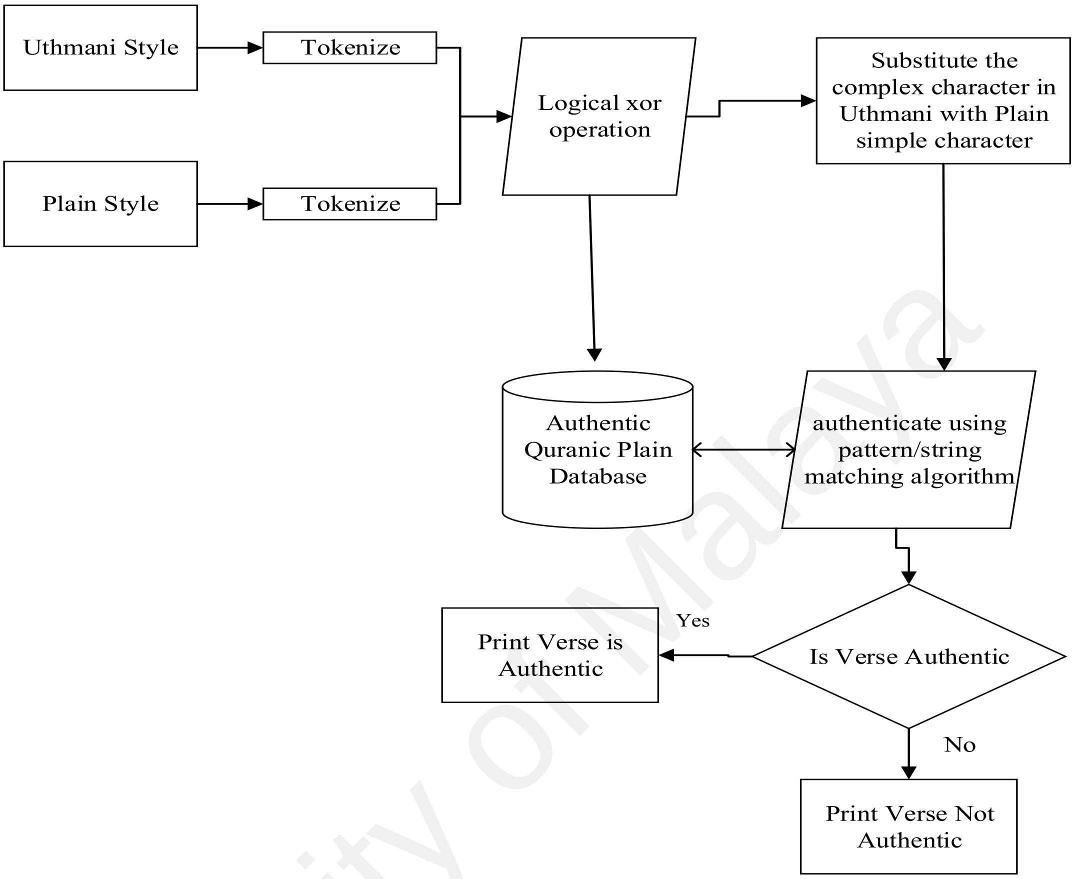

Uthmani Style
Tokenize
Plain Style
Tokenize
Logical xor operation
Substitute the complex character in Uthmani with Plain simple character
Authentic Quranic Plain Database
authenticate using pattern/string matching algorithm
Is Verse Authentic
Yes
Print Verse is Authentic
No
Print Verse Not Authentic

Figure 4.3: The Logical flow of the Residual approach for conversion of a given Quranic verse

The proposed approach is divided into four sub-sections. Firstly, tokenization of both the verses into segment components proposed in section (i). The residual is found using the logical XOR operation in section (ii); the conversion is done by substituting a suitable letter with the help of reference database in section (iii); the converted verse is verified using a string matching algorithm in section (iv).

(i) Tokenization for Segmenting Components from Input Verses

The most widely used encoding schemes for English texts is the American Standard Code for Information Interchange (ASCII). This encoding uses seven bits to represent a single English character (McEnery & Xiao, 2005), which suffices for simple scripts like English. However, for complex scripts like Arabic, it is not suitable as it requires more than seven bits for representation. Therefore, to handle complex text, generally, the UTF-16 encoding scheme is used because UTF-16 constitutes a variable length encoding (McEnery & Xiao, 2005) scheme which represents Arabic text with diacritical symbols considerably well. Samples of the Unicode values for Arabic letters are shown in Table 4.3.

Table 4.3: Sample UNICODE representation

Quranic Letters	UTF-16 Representation
	U+0627
	U+0628
	U+062A
	U+062B
	U+062C
	U+062D
	U+062E
	U+062F
	U+0630
	U+0631
	U+0632
	U+0633
	U+0634
	U+0635

The proposed approach uses Unicode for Arabic text when tokenizing components (characters) from a given verse. We explore the regular expression approach (Chang & Manning, 2014; Strötgen et al., 2014) for splitting verses into character components, as it provides the delimiter (“ ”), which splits a given string into character by character representation with the help of Unicode. More details of segmenting character components from verse can be found in (Chang & Manning, 2014). An example of a character component segmentation for a Uthmanic verse is shown in Table 4.4.

Table 4.4: Tokenized Quranic Verse

Uthmani Verse	Tokenized Verse
كن ر دحرث ٱل خرة ز فُ حرثۦ و كن ر د حرث ٱلتُّي ثۦ نه و فُ ٱل خرة صيب	م ن ك ن ي ر ي ح ر ث ٱلْ خ ر ة ن ز ل ُ ف ى ح ر ث ۦ و م ن ك ن ي ر ي ح ر ث ٱل ّ ن ي ن ؤ ت ۦم ن و م ل ُ ف ٱلوْ خ ر ة م ن ن ّ ص يب

(ii) XOR Operation for Residual

The segmented character components of the Uthmanic text are converted to its binary representation and compared to the binary representation of character components (of reference database) by using an XOR operation. The output of zero (0) indicates both inputs are consistent and correct (Mano, 2017; Tayan et al., 2014). However, the output of one (1) indicates the difference called ‘residual’ as shown in Table 4.5, where it can be observed that “1” marked by a red colour is representing the residual of the Uthmanic and the plain Quranic text.

In Table 4.5, the number of 1’s highlighted in red depicts the bitwise differences between the two input strings. The differed characters indicating bit

1, are retrieved by creating a function using "*for*" Loop, where based on an index of differing characters, the values of bits are returned. Finally, all major differences between the two writing styles are analysed using the proposed approach.

(iii) Substitution for Correction

In order to correct the difference explained by the previous step, and convert the Uthmanic style into plain style, the characters that had differed between the Uthmanic and plain text, were first identified. Table 4.6 shows the differences in writing characters in Uthmanic and plain-text. Next, the proposed approach determines the difference and consequently identifies the suitable symbol to substitute in place of the residual in order to restore the meaning of the Uthmanic characters.

Table 4.5: XOR operation of verses

Style	Verse	Tokenized Verse	Binary Bit Representation of Verse	XOR operation of verses
Uthmanic Style	ٰك ومٱدّ	لٰ ك ي و م ٱل ّ	00100000 11011001 10000101 11011001 10001110 11011001 10110000 11011001 10000100 11011001 10010000 11011001 10000011 11011001 10010000 00100000 11011001 10001010 11011001 10001110 11011001 10001000 11011001 10010010 11011001 10000101 11011001 10010000 00100000 11011001 10110001 11011001 10000100 11011000 10101111 11011001 10010001 11011001 10010000 11011001 10001010 11011001 10000110 11011001 10010000 00100000	0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
Plain Style	ك وم دّ	م ل ك ي و م ل	00100000 11011001 10000101 11011001 10001110 11011000 10100111 11011001 10000100 11011001 10010000 11011001 10000011 11011001 10010000 00100000 11011001 10001010 11011001 10001110 11011001 10001000 11011001 10010010 11011001 10000101 11011001 10010000 00100000 11011000 10100111 11011001 10000100 11011000 10101111 11011001 10010001 11011001 10010000 11011001 10001010 11011001 10000110 11011001 10010000 00100000	
Uthmanic Style	ٱح د ﷲربّ ٱ ٰ ي	لٱ ح م للّ ر بّ ٱل ع	00100000 11011001 10110001 11011001 10000100 11011001 10010010 11011000 10101101 11011001 10001110 11011001 10000101 11011001 10010010 11011000 10101111 11011001 10001111 00100000 11011001 10000100 11011001 10010000 11011001 10000100 11011001 10010001 11011001 10001110 11011001 10000111 11011001 10010000 00100000 11011000 10110001 11011001 10001110 11011000 10101000 11011001 10010001 11011001 10010000 00100000 11011001 10110001 11011001 10000100 11011001 10010010 11011000 10111001 11011001 10001110 11011001 10110000 11011001 10000100 11011001 10001110 11011001 10000101 11011001 10010000 11011001 10001010 11011001 10000110 11011001 10001110 00100000	0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
Plain Style	ح د ﷲربّ ي	ل ح م للّ ر بّ ل ع	00100000 11011000 10100111 11011001 10000100 11011001 10010010 11011000 10101101 11011001 10001110 11011001 10000101 11011001 10010010 11011000 10101111 11011001 10001111 00100000 11011001 10000100 11011001 10010000 11011001 10000100 11011001 10010001 11011001 10001110 11011001 10000111 11011001 10010000 00100000 11011000 10110001 11011001 10001110 11011000 10101000 11011001 10010001 11011001 10010000 00100000 11011000 10100111 11011001 10000100 11011001 10010010 11011000 10111001 11011001 10001110 11011000 10100111 11011001 10000100 11011001 10001110 11011001 10000101 11011001 10010000 11011001 10001010 11011001 10000110 11011001 10001110 00100000	

Table 4.6: Analysis of Uthmanic and plain Quranic verses

Uthmanic Arabic Text	Plain Arabic Text	Uthmanic Arabic Text	Plain Arabic Text
أ		ا ا	
		لّ	ل
ٱ		ا	ا
		ى	ي
و			
لّ	ل	ئ	ي
ى	ي	ا	و
او			ا
وٗ	و		

For instance, the changes made in Table 4.6 include the replacement of letters like different versions of the letter *alif* **“ٱ”, “** ٰ (Arabic subscript alef) **”** with a simpler one i.e. **“ ا”**. Similarly, letters like “ءا”, “د”, “ۆ”, were replaced by their simpler forms as shown in Table 4.6. The symbol **“ ّ ”** (*shadda*) is used to represent one letter twice (long consonant) during recitation (Fadi, 2017). Similarly, the symbol **“ۡ ”** (Arabic small high dotless head of خ is replaced by ْ (*sukoon*)). The purpose of placing a *sukoon* above or beneath the letter is to indicate no sound, while a dotless head of خsignifies the absence of a vowel. Some styles use sukoon, while some use the dotless خ. Hence, for improving the accuracy of authentication, the symbols including ۡ and ۡ were removed. Similarly, all forms of the letter *yaa* **“ى”** are substituted with a simpler form, i.e. **“ي”.** The other symbols that are removed to improve the detection accuracy are listed in Table 4.7. Removing these symbols does not alter the meaning (Alshareef & Saddik, 2012).

Table 4.7: Symbols removed to improve detection accuracy

Uthmanic/Plain Symbols	Uthmanic/Plain Symbols
ۓ (Yeh barree with hamza above)	ا (small high jeem)
م (Arabic small low meem)	ا (Arabic small high seen)
م (small high meem isolated form)	ا (Arabic small high lam alef)
م (Small high madda)	ا (Arabic small high meem initial form)
▬ (Arabic Tatweel)	۩(Arabic place of sajdah)
م (Arabic Small High Ligature Qaf With Lam With Alef Maksura)	ا (Arabic Small High Ligature Sad With Lam With Alef Maksura)

Samples of the symbols used and removed for substitution in plain text are now listed in Table 4.8.

Table 4.8: Pre-processing in BenchMark Dataset

Plain Symbols	Substituted by
أ	
أ	
ا ا	
ئ	
ى	
اْ	
ى	ي

(iv) *Exact Matching for Authentication*

In order to authenticate the converted input (done in the previous step), where Uthmanic and plain Quranic verses, were converted into a single format, we propose to use a pattern/string matching algorithm. For choosing an optimal exact matching algorithm, we analysed the performance (in terms of search time) of different character-based exact matching algorithms using datasets from tanzil.net, as shown in Table 4.9.

Table 4.9: Performance analysis of character-based exact matching algorithms (in milliseconds)

Verses tested	**BM algorithm**	**TBM algorithm**	**BMT algorithm**	**Quick Search algorithm (QS)**	**SSM algorithm**	**Horspool**
Time-complexity (Searching phase)	***O(mn)***	***O(n)***	***O(mn)***	***O(mn)***	***O(mn)***	***O(mn)***
رحٰ رحِيم	1246	1078	1056	**1006**	1017	1030
وٰم ر ذ لفـرو أن س وت و لأرضكت رتقً ففتق ه وجن	1053	1087	945	1029	**944**	1072
و ذ رك ذ لفـرو نت ذوك لا هزوً أٰهذ ذي لكر نقلكم و هم بـلكر	968	957	**918**	924	951	1008
س ر ه يهم س عي ل ق ية أ م حـرو ًفـترى قوفـي ه صر ا ك أ هم	938	948	**898**	948	901	933
فـك هةً وبـلًّ	875	874	870	862	**856**	892
و ا ذٰك يـد	949	877	**876**	940	889	896

From our experiments conducted in Table 4.9, it was observed that there is no clear winner from different variants of the Boyer-Moore's character-based algorithms. The algorithms that were tested include the Boyer-Moore algorithm (Boyer & Moore, 1977), the turbo Boyer-Moore algorithm (Crochemore, 1994), the tuned Boyer Moore algorithm(Hume & Sunday, 1991), the Horspool algorithm (Horspool, 1980a) and the SSM algorithm (Al-Ssulami, 2014). It can be observed from Table 4.9 that the Tuned Boyer-Moore (BMT) algorithm performed slightly better, compared to the other approaches. Hence, the BMT algorithm was applied for matching purposes.

However, to understand the methodology of the above algorithms, a good understanding of the Boyer-Moore (BM) algorithm is required. The Boyer-Moore algorithm (Boyer & Moore, 1977; Faro & Lecroq, 2013) begins searching characters from right to left of a given pattern. In case of a mismatch, it shifts as many as *m* characters as shown in Figure 4.4 (where *m* denotes the length of patterns to be searched and n denotes the length of a given text).

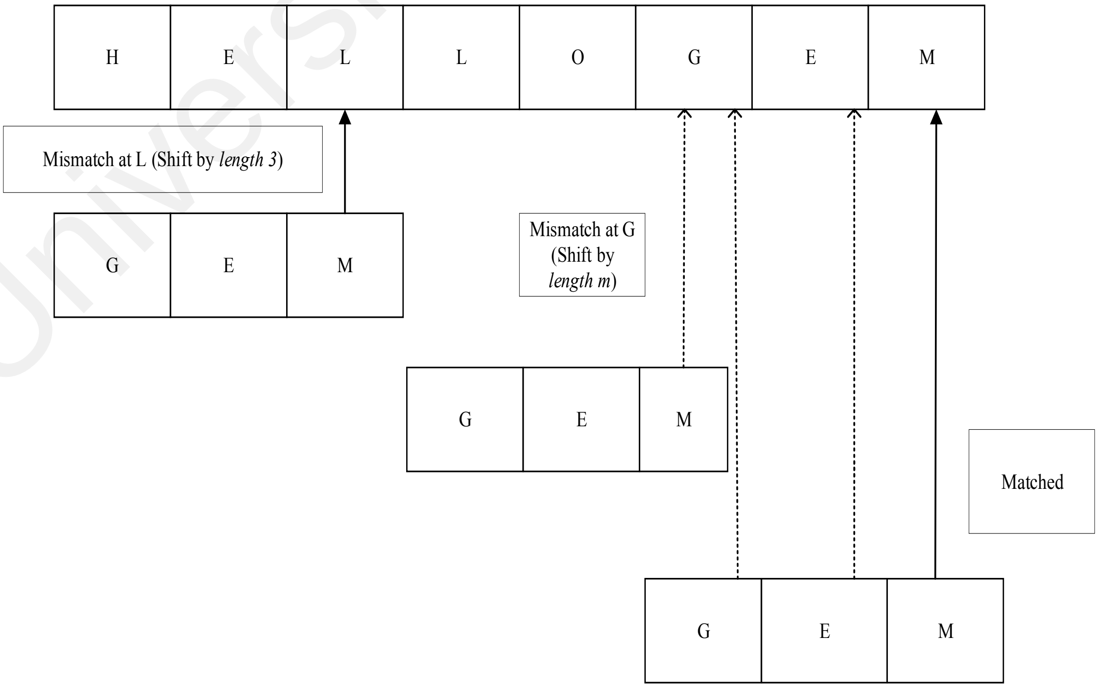

H
E
L
L
O
G
E
M
Mismatch at L (Shift by *length 3*)
G
E
M
Mismatch at G (Shift by *length m*)
G
E
M
Matched
G
E
M

Figure 4.4: The Boyer Moore Algorithm

The algorithmic steps are as follows:

- Search for a given pattern from the right side of the window and use the bad match rule to skip characters in case of a mismatch.

  Pre-processing: In this stage, a table is created, which gives values indicating the degree of shifting required in case of a mismatch (bad-match table). Once a character mismatch occurs, the algorithm shifts characters to the right side of the pattern either by *m* places or based on the position of the mismatching character (of a given text that has been matched before in a given pattern).

  Searching starts from the tail of the pattern, i.e. from the right side to the left side of the text as compared to the naive algorithm where searching starts from the left. The algorithm works by computing the length of the search string and storing its value as a default shift length.

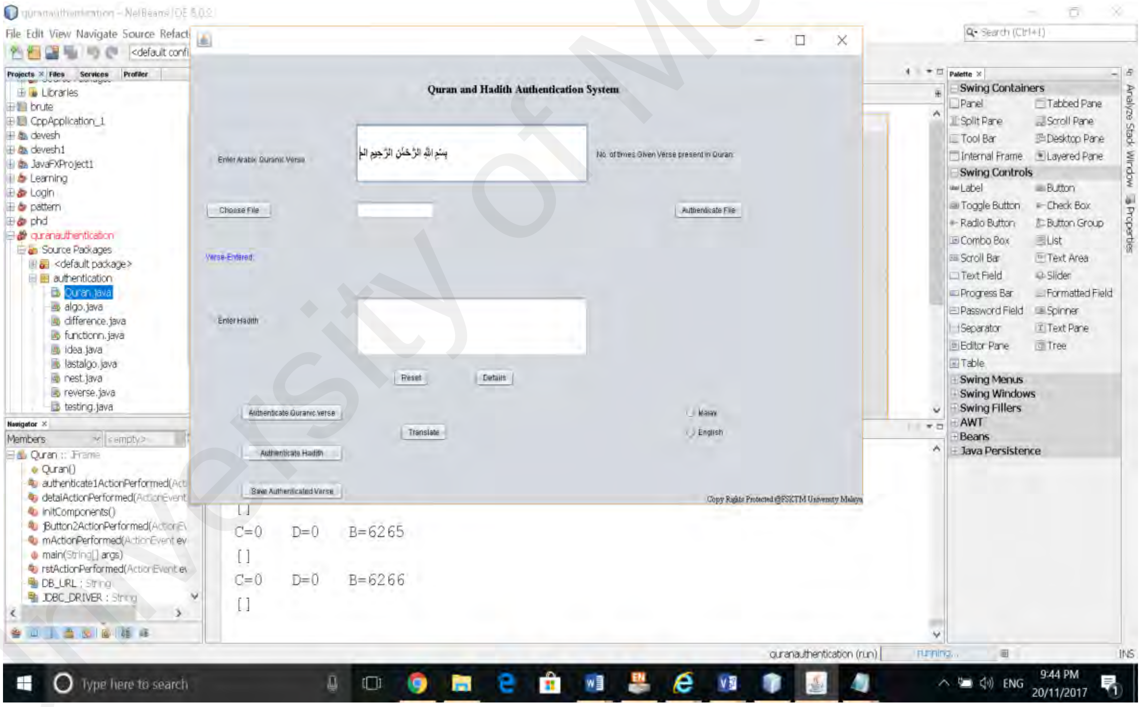
The algorithm has a time complexity of *O (n+m)* in the best case, and *O (n*m)* in the worst-case. Here, *m* denotes the length of pattern and *n* denotes the length of text that is to be searched.

4.4 Experimental results

To evaluate the proposed approach, we consider the authentic version of a Quran dataset available at Tanzil.net and our verified dataset (as mentioned in chapter 3), which has been used in previous research studies for determining the authenticity of Quranic texts. Tanzil.net has six types of writing, which include Uthmani, Simple, Simple Enhanced, Simple Minimal, Simple Clean and Uthmani Minimal (tanzil.net, 2016). In this work, we consider a Simple dataset as it contains all diacritics that are necessary to recite the Quranic text accurately. Besides, it consists of fewer symbols, which reduces the number of computations for verse verification. The pre-processed and verified dataset (by the experts from the Faculty of Islamic Studies, University of Malaya) is available on the website (http://quranhadith.fsktm.um.edu.my/).

4.4.1 Experiments for Authentication

The prototype of the proposed approach is shown in Figure 4.5, illustrating how the proposed method finds residuals and the correct verses. The system details for conducting our experiments include; Java with IDE Netbeans 8.02. The hardware used was an i-5 Intel Processor, 4 MB cache and a 4 GB RAM with a Windows 10 Operating system. We randomly chose 1000 Quranic Uthmanic verses from the database to measure the performance.



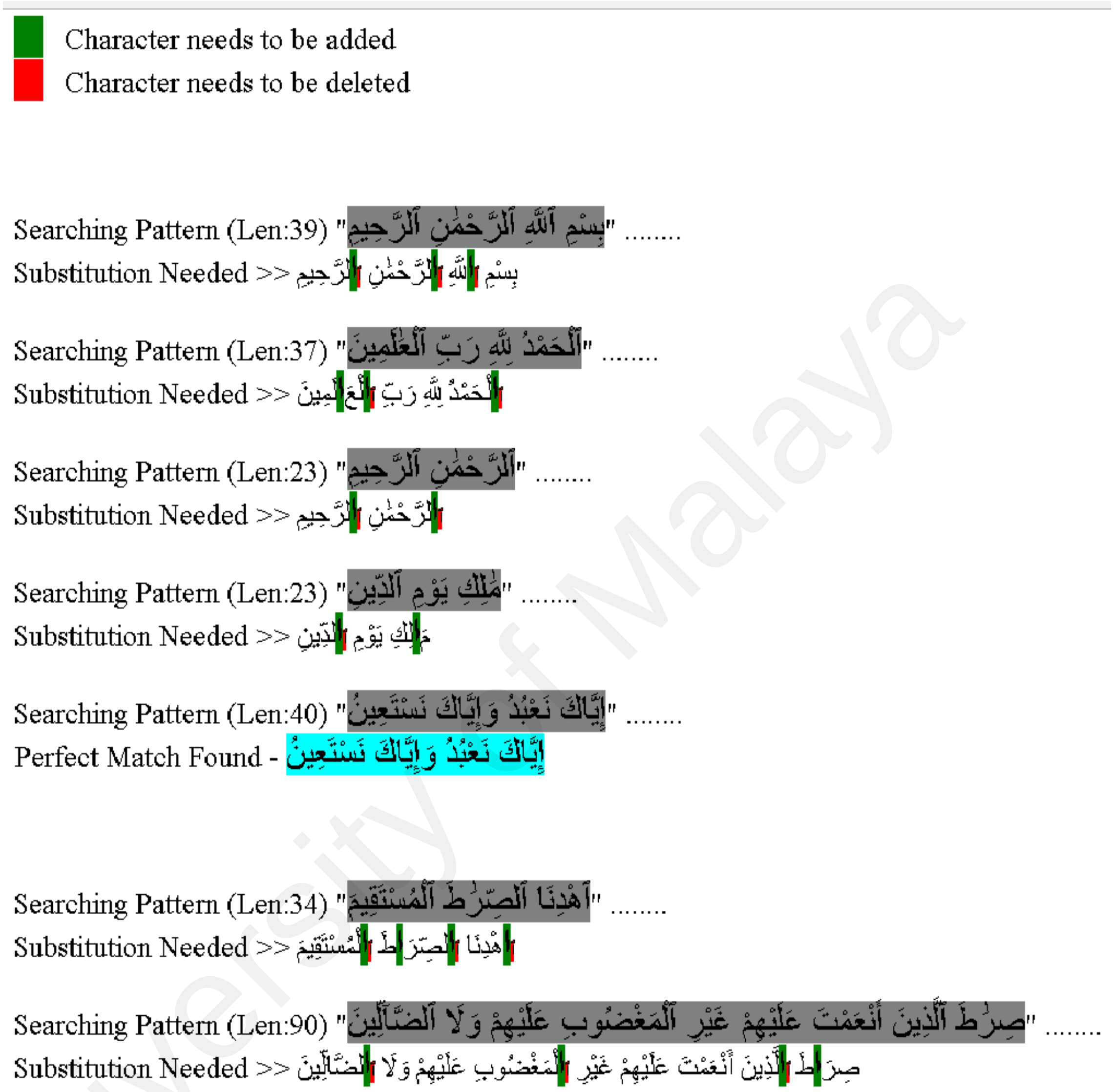
Quran and Hadith Authentication System

Enter Arabic Quranic Verse: بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ الم

```
[]
C=0     D=0     B=6265
[]
C=0     D=0     B=6266
[]
```

Figure 4.5: Prototype of Quran authentication system

The conversion of the first chapter of the Holy Quran (i.e. Surah Fatihah) written in a Uthmanic style to Plain Quranic style is shown in Figure 4.6. The characters highlighted in red color indicates, there was the output of one (1) during residual operation and these characters need to be analyzed for substitution. Characters highlighted in green are then substituted based on Table 4.6 and 4.8 respectively.



🟩 Character needs to be added
🟥 Character needs to be deleted

```
Searching Pattern (Len:39) "بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ" ........
Substitution Needed >> بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Searching Pattern (Len:37) "ٱلْحَمْدُ لِلَّهِ رَبِّ ٱلْعَٰلَمِينَ" ........
Substitution Needed >> الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ

Searching Pattern (Len:23) "ٱلرَّحْمَٰنِ ٱلرَّحِيمِ" ........
Substitution Needed >> الرَّحْمَٰنِ الرَّحِيمِ

Searching Pattern (Len:23) "مَٰلِكِ يَوْمِ ٱلدِّينِ" ........
Substitution Needed >> مَالِكِ يَوْمِ الدِّينِ

Searching Pattern (Len:40) "إِيَّاكَ نَعْبُدُ وَإِيَّاكَ نَسْتَعِينُ" ........
Perfect Match Found - إِيَّاكَ نَعْبُدُ وَإِيَّاكَ نَسْتَعِينُ


Searching Pattern (Len:34) "ٱهْدِنَا ٱلصِّرَٰطَ ٱلْمُسْتَقِيمَ" ........
Substitution Needed >> اهْدِنَا الصِّرَاطَ الْمُسْتَقِيمَ

Searching Pattern (Len:90) "صِرَٰطَ ٱلَّذِينَ أَنْعَمْتَ عَلَيْهِمْ غَيْرِ ٱلْمَغْضُوبِ عَلَيْهِمْ وَلَا ٱلضَّآلِّينَ" ........
Substitution Needed >> صِرَاطَ الَّذِينَ أَنْعَمْتَ عَلَيْهِمْ غَيْرِ الْمَغْضُوبِ عَلَيْهِمْ وَلَا الضَّالِّينَ
```

Figure 4.6: Conversion of Input to one single format

The proposed approach authenticated 871 verses out of 1000 verses of the Digital Quran. The experiments were done on small, medium and long chapters of Digital Quran. The verses were selected randomly from different chapters of Digital Quran. The preprocessed data set was evaluated by the Faculty of Islamic Studies, the University of Malaya for the correctness.

$$Accuracy = \frac{\text{Number of particular verses Found}}{\text{Total number of particular verses}} \tag{4.1}$$

Thus, Accuracy = 871/1000 = 87.1%.

4.4.2 Effectiveness of the Proposed Approach

To show the usefulness of the proposed approach which converts the verse by substituting suitable symbols at their residual locations, we conducted experiments by feeding input directly into the BMT algorithm and authenticate without correction as shown Table 4.10. Basically, the algorithm checks whether the Uthmani verses can be authenticated in the Plain Quran dataset. Table 4.10 shows that the BMT algorithm was unable to detect the verses due to different arrangements of diacritics in the Uthmani and the Plain dataset. However, when the corrected verse given by the proposed approach for the BMT algorithm is used as input, the same verses are shown in Table 4.10 were authenticated correctly. Thus, the proposed conversion by substitution proved useful and effective.

Table 4.10: Analysis of Quranic styles without using XOR and Substitution

Input Verse	BMT algorithm	Proposed approach	Benchmark dataset
مِنَ ٱلۡجِنَّةِ وَٱلنَّاسِ	Not Able to verify	مِنَ الْجِنَّةِ وَالنَّاسِ	مِنَ الْجِنَّةِ وَالنَّاسِ
فَذَٰلِكَ ٱلَّذِي يَدُعُّ ٱلۡيَتِيمَ	Not Able to verify	فَذَٰلِكَ الَّذِي يَدُعُّ الْيَتِيمَ	فَذَٰلِكَ الَّذِي يَدُعُّ الْيَتِيمَ
مِن شَرِّ مَا خَلَقَ	Not Able to verify	مِنْ شَرِّ مَا خَلَقَ	مِنْ شَرِّ مَا خَلَقَ

4.4.3 Comparative Study

To show the superiority of the proposed approach, we compare the results of our proposed approach with the other existing approaches. Firstly, we consider the Quran

Quote Verification Algorithm (QQV), which removes all diacritics from the input Quranic verse and verifies the authenticity by using the dataset (Alshareef & Saddik, 2012). Secondly, we consider the Quran Verification and Authentication Algorithm which encodes input using the UTF encoding scheme and verifies it using the UTF-based dataset (Alginahi et al., 2013). Finally, we compare our approach with the hashing algorithm, which generates a hash value using existing algorithms like MD5. Thereafter, the authenticity is verified based on the hash values from the given dataset (Alsmadi & Zarour, 2015). The sampled qualitative and quantitative results of the proposed approach and the other existing approaches are shown in Table 4.11, where it is observed that all the existing approaches failed to authenticate due to a mismatch between the Uthmanic verse input and the Plain Quran verse input. This is valid because both verses differ in their arrangement of the diacritics. Therefore, the accuracy of the authentication of the existing approaches is 0.0%, while the proposed approach method achieved 87.1% accuracy. It corrects the mismatch between the Uthmani verse and the plain Quran verse through residual searches and substitution, as shown in Table 4.11.

Table 4.11: Comparative Analysis after XOR and Substitution Phase

Example	**Quran Quote Verification Algorithm (QQV) (Alshareef & Saddik, 2012)**	**Quran verse Verification and authentication Algorithm(Alginahi et al., 2013)**	**Hashing Algorithm (Alsmadi & Zarour, 2015)**	**Proposed Approach**	**Benchmark Verse**
Input	Output	Output	Output	Output	
ٱ جنة وٱن س	ٱ جنة وٱن س	ٱ جنة وٱن س	1192663878	جنة و ن س	جنة و ن س (- 1681347706)
فـي دو رب هٰذ ٱيت	فـي دو رب هذ يت	فـي دو رب هذ يت	-725617782	فـي دو رب هذ يت	فـي دو رب هذ يت (604767505)
Accuracy	0.0 %	0.0 %	0.0 %	87.1 %	

The proposed approach does not work well for the verses which contain extra characters. For example, the following Uthmanic verse “ ءَأَنتُم ٰ أ زت و ٱ زن أم ح ٱ نز ون ” starts with a letter “ ” and the plain verse “ أَأَنْتُمْ أ زت و زن أم ح نز ون ” starts with a letter “أَ”. Since both the verses are correct, but there is no substitution possible for these kinds of verses. In case, a letter “ ” is substituted with the letter “أَ”, then the remaining verses of Digital Quran containing a letter “ ” will also change resulting in a more severe problem. Similarly, the following plain verse “ يَسْأَلُه ٰ ف س و ت و لأرض كل وم هو ف شأن ”, contains extra alif in word “يَسْأَلُه”. In Uthmanic version, the word “ سً ُ” does not contain any extra alif ( ) and the letters “ ” and “ ” are connected directly. This results in a mismatch. Those types of words result in lower accuracy. A few other samples for which the proposed approach does not perform well, are listed in Table 4.12, where the difference is highlighted. From Table 4.12, it can be observed that there are cases (for example serial no. 1,2 7,8), where some extra characters like *alif* are embedded in Uthmanic text compared to reference database making conversion process inevitable. Similarly, in serial no. 9, the character “ ” in plain text style is represented by “ ″ in Uthmanic style. In these cases, the substitution method is not feasible considering the sensitive nature of the Quran. Therefore, there is scope for extension of the proposed work to find a solution to the above-mentioned issue.

Table 4.12: Unverified verses using the proposed approach

Serial no.	Uthmanic	Plain
1	و شرّ ٱفثتف ٱ قد	و شرّ فث تف قد
2	قل أُٰهّ ٱلفٰرون	قل أهّ لفرون
3	ولا أ ْ بد نتّم	ولا أ بد نتم
4	أ طيٰك ٱلكوثر	أ طي ك لكوثر
5	ولا حضُّ ٰ ط م ٱ سلمي	ولا حضُّ ط م سلمي
7	لإ ٰفقر ش	لإ لافقر ش
8	فَ هم رح ة ٱ تّ وٱصيف	لفهم رح ة تّ و صيف
9	ٱ ذى أط هم ّ جوع و ن هم ّ خوف	ذي أط هم جوع و ن هم خوف
10	أ م ترلكيفف ل ربُّك بأصحٰب ٱلفيل	أ م ترلكيفف ل ربُّك بأصح ب فيل

4.5 Summary

In this chapter, the first phase of authenticating diacritical sensitive texts involving digital Quran verses is discussed. A new approach is proposed to convert multi-script styled texts into a common format for authentication purposes. The proposed approach determines the residual between the input verse and the plain text through an XOR operation. The proposed approach examines the residual in order to identify the suitable symbol and substitutes the error symbol where the Uthmani script letter differs from the plain script. Subsequently, the corrected version is validated by using the BMT algorithm. The experimental results show that the proposed approach achieves 87.1% authentication accuracy and outperforms the existing approaches in terms of accuracy. The existing approaches do not perform as well since their suitability is limited to a single type of script style.

The next phase focuses on enhancing the retrieval process of single diacritical text in the authentication phase that takes a considerable amount of time to verify the authenticity of texts. The details of the authenticating single DHQ text are discussed in Chapter 5.

CHAPTER 5: AN EFFICIENT DATA REPRESENTATION FOR SEARCHING AND RETRIEVING DIACRITIC ARABIC TEXT

5.1 Introduction

Chapter 4 discusses the preprocessing phase in which the different script formats are converted into a standard format. This chapter outlines the second phase of this research, the authentication phase. Diacritical text like the Quranic text is usually not retrieved in the most efficient manner due to inefficient data representation issues (discussed in Chapter 2). This results in more time needed to generate authentication results. The factors like searching algorithm, database, input length and size that affect the retrieval and search process are further discussed using the factorial design approach. To improve the retrieval and search process, a new method is proposed that tokenizes the given input and retrieves it by identifying the first character of the verse. The shortened search time makes the retrieval process more efficient. The related studies are mentioned in Section 5.2, followed by the proposed approach in Section 5.3 and the discussion of the results in Section 5.4, and the conclusion in Section 5.5.

5.2 Related work

As data storage and processing increases, data representation and design also require changes in order to cope with the challenges of complex databases, with heterogeneous data. There are numerous methods available in the literature that focus on new methods for improving time complexity and desired results, but those approaches hardly focus on the design of such databases. The Non-Latin diacritical texts like Digital Quran and Al-Hadith (teaching of Prophet Mohammad (peace be upon Him)) content are being uploaded on the Internet through social media websites, blogs, etc every second without organizing data in a particular way that causes inefficient retrieval. (Mohammed et al., 2015) identified that errors found in the uploaded content were due to missing diacritic symbols. In the case of diacritical texts, the position of diacritic symbols is vital for correctly reading and understanding the meaning of the whole sentence (for example, the Quran verse). This shows that there is a need for better representation and retrieving methods which represent data without errors and missing symbols. Moreover, many search engines such as Google, Yahoo, and MSN are not efficient enough for retrieving non-Latin texts involving diacritics from databases. Such engines are good for English texts that use the ASCII encoding scheme and involves 8 bits/character but are not efficient for non-Latin scripts that use multi-bytes for representing a single character (Bar-Ilan & Gutman, 2005; Hakak et al., 2017c). It appears that good representations for database designs are at an infant stage, especially for diacritical texts, such as Arabic, Urdu and Farsi etc (Al-Badarneh et al., 2016; Al-Sanabani & Al-Hagree, 2016). Thus, there is immense scope for proposing new representations for the above mentioned digital texts, in order to enable such search engines to efficiently and precisely retrieve the required data in real time.

Several methods (Al-Badarneh et al., 2016; Hammo, 2008) were found in the literature for representing data in English texts. However, those methods may not be used directly to represent diacritical scriptures. The reason is that diacritical scripts are sensitive compared to English data and require four stages of pre-processing, indexing, querying, and finally, retrieving (Atwan et al., 2015). Moreover, it is sensitive in nature, as for example, given that the position of an isolated dot changes the meaning of the whole sentence (verse), while in the case of English, changing one character does not significantly affect the meaning of the sentence. Therefore, diacritical scripts require an accurate and efficient representation (Atwan et al., 2015).

Arabic is one of the most influential and widely spoken languages, with approximately 350 million native speakers (Al-Badarneh et al., 2016; Al-Sanabani & Al-Hagree, 2016; Khalaf et al., 2014). It comes under the family of Semitic languages and differs syntactically and morphologically with Latin languages. Arabic is written from right to left and has 23 consonants with three long vowels. The vowels used in Arabic are short and popularly known as diacritics. Those diacritics are written either above or beneath the consonant to give the word a desired sound and meaning. Native speakers usually do not require diacritics for reading or understanding Arabic text for daily activities like reading magazines, textbooks, letters and so on. However, the use of diacritics is heavily prescribed and recommended for religious scriptures. In the Arabic language, the two most sensitive and most important religious scriptures include the Quran (the holy book of Muslims) and Al-Hadith (teachings of Prophet Mohammad (PBUH). For example, Table 5.1 shows details of diacritics used in Quran with the representation given using the UTF-16 encoding scheme. Table 5.2 shows that the positions of the symbols are context sensitive.

Table 5.1: Description of Diacritics used in Arabic texts (Sabbah & Selamat, 2013)

Diacritics used in Arabic texts	Description	Symbols (Alshareef & Saddik, 2012)	Sound (Ryding, 2005)	UTF-16 Representation
Fatha	Small diagonal line above a letter	َ	aa	U+064E
Kasra	Small diagonal line below a letter	ِ	ai	U+0650
Damma	Small “comma-like” diacritic placed above a letter	ُ	au	U+064F
Tanwin	Double vowel diacritic at the end of verses	ً ٍ ٌ	Ain, aan, aun	U+064B U+064C, U+064D
Sukun	A small circle shape above the letter indicating that the consonant is not followed by a vowel		d	U+0652
Shadda	A Small circle shape above the letter indicating that the consonant is not followed by a vowel	ّ	dd	U+0651
Madda	Diacritic appears on top of alif indicating a long alif		aa	U+0622

For example, an Arabic word كتب consisting of three consonants i.e. ب ت ك gives different meanings when a different arrangement of diacritics are used (Kirchhoff & Vergyri, 2005). More details are shown in Table 5.2.

Table 5.2: Different interpretations of the word كتب with different diacritics (Hammo, 2008)

Arabic Word	Transliteration	Part of Speech	Meaning (in English)
كتب	kataba	Verb	Wrote
كتب	kutub	Noun	Books
كتب	kutiba	Verb	Written
كتب	kattaba	Verb	Make someone to write

In summary, the above discussion shows that diacritical scripts are sensitive and hence, requires an accurate representation to retrieve data accurately with their correct meanings. In addition, good representation results in efficient retrieval.

As discussed previously, less attention has been paid towards the database organization and representation of diacritical texts compared to the new methods which generally explore different string/pattern matching methods to achieve efficiency. We review the methods related to data representation and pattern matching. For example, standard and popular pattern matching algorithms include Boyer-Moore, BMT, KMP and Rabin Karp (Faro & Lecroq, 2013). Those algorithms have been enhanced to improve the search time and accuracy (Faro & Lecroq, 2013; Hlayel & Hnaif, 2014). Furthermore, such algorithms are limited to Latin texts only and may not be suitable for non-Latin texts like Arabic without pre-processing (Hlayel & Hnaif, 2014). (Alsmadi & Zarour, 2015) proposed an algorithm for searching and verifying of Arabic Quran verses. The proposed algorithm is based on the hashing approach and involves removal of diacritics which makes the verification process questionable. In Arabic religious scripts, diacritics are vital. Alginahi et. al (Alginahi et al., 2013) proposed an algorithm for detecting Quranic Arabic text from websites. This approach also removes diacritics to achieve their objectives. Therefore, the retrieved data cannot be properly authenticated due to the removal of diacritics. In the same area, Sabbah et.al (Sabbah & Selamat, 2013) and Alshareef et.al (Alshareef & Saddik, 2012) also proposed methods for retrieving text based on diacritics removal. However, (Al-Sanabani & Al-Hagree, 2016) proposed a method which does not focus on removal of diacritics for retrieving text from the database. This method considers the non-diacritic text for experimentation.

Similarly, there are search engines available related to searching Quranic verses online (Alshareef & Saddik, 2012). For instance, tanzil.net(tanzil.net, 2016), search-truth (Muslim-web, 2018; Searchtruth.com, 2018) and Muslim-web (Muslim-web, 2018) are some examples of search engines. Such search engines operate on full-words, as well as stem and word synonyms for retrieving text from the database. It is noted from our initial analysis, that these engines were not efficient enough for retrieving query for DHQ that involves diacritical text. When applying variable length verses, the performance of those search engines degrades, and in some cases, are unable to retrieve the requested verse. In addition, those three search engines require more time for verses with complex diacritics. Results based on different diacritical verses for some popular Quran search engines are shown in Table 5.3. The main reason for retrieving some verses while not retrieving others is most probably due to inefficient data representation that leads to inefficient retrieval. Pattern matching algorithms will be able to retrieve data more accurately and efficiently if the data organization is improved.

Table 5.3: Experiments on standard Quranic search engines

Chapter number (Surah)	Verses	Tanzil.net	Muslim-Web	Search truth
2	فَقُلْنَا اضْرِبُوهُ بِبَعْضِهَا ۚ كَذَٰلِكَ يُحْيِي اللَّهُ الْمَوْتَىٰ وَيُرِيكُمْ آيَاتِهِ	Retrieved	Retrieved	Retrieved
2	فَقُلْنَا اضْرِبُوهُ بِبَعْضِهَا ۚ كَذَٰلِكَ يُحْيِي اللَّهُ الْمَوْتَىٰ وَيُرِيكُمْ آيَاتِهِ لَعَلَّكُمْ	No Results	No Results	Retrieved
2	الم (١) ذَٰلِكَ الْكِتَابُ لَا رَيْبَ ۛ فِيهِ ۛ هُدًى لِلْمُتَّقِينَ (٢) الَّذِينَ يُؤْمِنُونَ بِالْغَيْبِ وَيُقِيمُونَ الصَّلَاةَ وَمِمَّا رَزَقْنَاهُمْ يُنْفِقُونَ (٣) وَالَّذِينَ يُؤْمِنُونَ بِمَا أُنْزِلَ إِلَيْكَ وَمَا أُنْزِلَ مِنْ قَبْلِكَ وَبِالْآخِرَةِ هُمْ يُوقِنُونَ (٤)	No Results	No Results	No Results
67	قُلْ هُوَ الرَّحْمَٰنُ آمَنَّا بِهِ وَعَلَيْهِ تَوَكَّلْنَا ۖ فَسَتَعْلَمُونَ مَنْ هُوَ فِي ضَلَالٍ مُبِينٍ (٢٩)قُلْ أَرَأَيْتُمْ إِنْ أَصْبَحَ مَاؤُكُمْ غَوْرًا فَمَنْ يَأْتِيكُمْ بِمَاءٍ مَعِينٍ (٣٠)	No Results	No Results	No Results
114	قُلْ أَعُوذُ بِرَبِّ النَّاسِ (١) مَلِكِ النَّاسِ (٢) إِلَٰهِ النَّاسِ (٣) مِنْ شَرِّ الْوَسْوَاسِ الْخَنَّاسِ (٤) الَّذِي يُوَسْوِسُ فِي صُدُورِ النَّاسِ	No Results	No Results	No Results

The motivation for the proposed data representation is due to the fact that the Unicode scheme needs more bytes to process and read data. The approach in this study employs a UTF-16 encoding scheme. The example below will clarify the need for our new data representation scheme for diacritical texts.

Suppose the input Quranic verse that needs to be retrieved and searched is “ الرَّحْمَنِ الرَّحِيمِ’. The verse consists of 22 characters, including diacritics and each character takes more than one byte of storage to process. This implies that in order to retrieve and search the given verse, the header needs to traverse through the whole Quran database and match the required 1408 bits. While traversing through the database, there is a possibility of garbled values appearing within the database during the writing operation. This affects the retrieval process and increases the searching process as showed through factorial experiments below.

However, in the case of English texts, the ASCII encoding scheme is used. As mentioned above, ASCII usually takes 7 bits to process one character. This implies for 22 characters; ASCII scheme will take around 154 bits for the header to read and retrieve the results. Hence, to retrieve and search English texts, it takes less time compared to diacritical Arabic texts and is efficiently retrieved in most of the cases. Experiments were conducted in this study where it was found that the encoding had affected the retrieval and searching process as shown in Table 5.4. Two well-known string matching algorithms were taken, i.e. Boyer-Moore (BM) (Boyer & Moore, 1977) and SSM (Al-Ssulami, 2014). Both diacritical and English texts of similar length were taken. It was observed that diacritical-based texts take more time as compared to English texts. The reason was based on the above-mentioned observation.

Table 5.4: Effect of encoding schemes on the retrieval and searching process

Algorithm	**Arabic texts (UTF Encoding)**	**Length of Text**	**Pattern Searched**	**Time (in milliseconds)**	**English texts (ASCII Encoding)**	**Length of Text**	**Pattern Searched**	**Time (in milliseconds)**
BM	صرٰط ٱلذ أ ت يهم غير ٱ غضوب يهم ولا ٱضٓي	90	لا	**12.6**	Thus, the heavens and the earth were finished and all the host of them on the seventh day God	90	day	**5.9**
SSM	صرٰط ٱلذ أ ت يهم غير ٱ غضوب يهم ولا ٱضٓي	90	لا	**13.9**	Thus, the heavens and the earth were finished and all the host of them on the seventh day God	90	day	**6.8**

Based on the above discussion, one can confirm that most of the existing methods focus on new approaches for pattern searching to achieve time efficiency. On the contrary, only a little attention has been given to improving efficient data representation and organization, which also contributes to improving time efficiency (Hammo et al., 2007;

Nisha et al., 2014). As highlighted previously, there are methods that focus on data representation and organization. However, such methods only work well by either removing diacritics or considering non-diacritical texts. This task involves an overhead to authenticate the retrieved data to confirm the meanings. Although search algorithms can be enhanced to improve retrieval and searching process, thereby improving authentication process, it would be interesting to know the effect of the database (involving different datasets) on the overall performance of search algorithm. This can be achieved by a factorial design approach.

5.2.1 Factorial Design

It is noted from the studies of (Hakak, 2014), that there are certain factors that can degrade the performance of an algorithm and factorial design is such an approach that can help to determine the effect of each such factor. While searching some text, usually three important factors i.e. searching algorithms (like pattern matching algorithm), the type of datasets (like protein, genome) and size of input are required. To determine the influence of factors like the selection of search algorithm, type of datasets and size of input on retrieval process can help us to know, whether an efficient need of database (where datasets are stored) is needed or not. The aim is to determine whether the type of database (for example protein and genome) has any influence on retrieval process and there are many kinds of factorial models available for evaluating the effect of a particular factor on a response variable.

The fundamental purpose of a factorial model is to get and provide the maximum amount of information through the minimum number of experiments. The most widely used factorial designs are Simple Designs, Full factorial Designs, and Fractional Designs. Simple Design is a very straightforward technique in which only one factor is varied and at a time effect of only one factor can be evaluated against a response variable. Although this design is simple, statistically, it is not efficient due to the fact that if some factors have interaction with each other, using this model may lead us towards wrong conclusions. Fractional Factorial Design is used in complex research studies involving rigorous and too many experiments. The advantage of fractional factorial design method is that it saves time and cost as compared to the full factorial design method. However, the flaw of this technique is that it does not give much detailed information which a full factorial design technique gives (Jain, 1991).

Full Factorial Design is an extension to the simple design model and utilizes all possible combinations of all levels of all factors. One of the major advantages of full factorial design method is that all possible combinations of configuration are examined under the same workload. The effect of every factor can be found along with interactions among the factors. However, there is one flaw in this experimental design and that is the time and cost of the study. In the full factorial model, the $2k$ factorial design is the most popular and widely used technique involving many studies, like in chemical industry, management industry, material engineering and so on. The $2k$ Factorial Design Model is used to find the effect of k factors with each k factor having two possible levels or values. This model is most widely used due to its easiness to analyze things and sorting out factors in a particular order. By $2k$, it means there are k factors with two levels and total experiments are performed $2k$ times. After performing the experiment $2k$ times, we get $2k$ effects which include k main effects, two-factor interactions, three-factor interactions and so on. We propose to use the $2k$ model also due to its usefulness compared to the above-mentioned models. In this model, three factors are taken, and each factor is assigned two levels denoted by 1 and -1. Once each factor is assigned two levels below 1 and -1, then the following formula is used to find the effect of factors:

$$SST = 2^3\left(q_A{}^2 + q_B{}^2 + q_C{}^2 + q_{AB}{}^2 + q_{AC}{}^2 + q_{BC}{}^2 + q_{ABC}{}^2\right) \quad (5.1)$$

Here A, B, C denotes different factors AB, BC and AC denote the interaction between the two factors and ABC denotes interaction among all three factors. SST denotes the sum of square total (Jain, 1991).

To calculate SST, we need to consider the following steps to measure the effect of the above three factors i.e. pattern matching algorithm, type of datasets and size of the input.

- *Data-collection Phase:*

  We collect a dataset that is used in Faro (Faro & Lecroq, 2013) which includes protein and genome data sets. We consider two classic string/pattern matching algorithms of backward oracle matching (BOM) and extended backward oracle matching (EBOM) algorithms with the input data size of 2MB and 1024MB respectively. The reason to choose these two algorithms is that both algorithms have the same structure with EBOM having a faster loop. However, one can choose any algorithms to check the effect on the retrieval process. Our intention here is to show an objective analysis for estimating the effect of three factors.

Table 5.5: Three factors for Designing Factorial Model

	Throughput	Levels	
		-1	1
A	Pattern Matching Algorithm	BOM	EBOM
B	Type of database	Genome	Protein
C	Input Size	2 MB	1024 MB

- *Factorial design Phase:*

  In this phase, the 2k factorial Design Model was designed based on factors shown in Table 5.5. For more details, one can refer to (Faro & Lecroq, 2013). *A* denotes pattern matching factor i.e. search algorithm, *B* denotes the dataset/Database type factor and *C* denotes Input size factor. All these factors have two levels, as shown in Table 5.6.

Table 5.6: 2K Factorial Design Results

A	B (-1)		B (1)	
	Database Type (Genome Sequence Database)		Database Type (Protein Sequence Database)	
	C		C	
Pattern Matching Algorithm	Input Size	Input Size	Input Size	Input Size
	-1	1	-1	1
	2MB	1024MB	2MB	1024MB
BOM (1)	136.3	2.05	26.49	0.73
EBOM (-1)	49.77	3.42	12.13	2.57

The effect of parameters can be calculated using equation (5.1). The values of A(q_A) , B(q_B - - - up to ABC(q_{ABC}) are calculated by multiplying Boolean values of A , B , C up to ABC with column *Y* (In *Y*, the experimental results of Table 5.6 are presented) given in Table 5.7. To calculate SST values, the total values of A, B, C, AB, AC, BC, and ABC are divided by 8.

$$q_A = (-1*49.77) + (1*136.3) + (-1*12.13) + (1*26.49) + (-1* 3.42) + (1* 2.05) + (-1*2.57) + (1*0.73) = \mathbf{97.68/8 = 12.12} \quad (5.2)$$

$$q_B = (-1*49.77) + (-1*136.3) + (1*12.13) + (1*26.49) + (-1 *3.42) + (-1* 2.05) + (1*2.57) + (1*0.73) = \mathbf{-149.62/8 = -18.7025} \quad (5.3)$$

$$\begin{aligned} q_C = {} & (-1*49.77) + (-1*136.3) + (-1*12.13) + (-1*26.49) + (1 \\ & *3.42) + (1*\ 2.05) + (1*2.57) + (1*0.73) \\ & = \mathbf{-215.92/8} = \mathbf{-26.99} \end{aligned} \tag{5.4}$$

$$\begin{aligned} q_{AB} = {} & (1*49.77) + (-1*136.3) + (-1*12.13) + (1*26.49) + (1 \\ & *3.42) + (-1*\ 2.05) + (-1*2.57) + (1*0.73) \\ & = \mathbf{-72.64/8} = \mathbf{-9.08} \end{aligned} \tag{5.5}$$

$$\begin{aligned} q_{BC} = {} & (1*49.77) + (1*136.3) + (-1*12.13) + (-1*26.49) + (-1 \\ & *3.42) + (-1*\ 2.05) + (1*2.57) + (1*0.73) \\ & = \mathbf{145.28/8} = \mathbf{18.6} \end{aligned} \tag{5.6}$$

$$\begin{aligned} q_{AC} = {} & (1*49.77) + (-1*136.3) + (1*12.13) + (-1*26.49) + (-1 \\ & *3.42) + (1*\ 2.05) + (-1*2.57) + (1*0.73) \\ & = \mathbf{-104.1/8} = \mathbf{-13.0125} \end{aligned} \tag{5.7}$$

$$\begin{aligned} q_{ABC} = {} & (-1*49.77) + (1*136.3) + (1*12.13) + (-1*26.49) + (1 \\ & *3.42) + (-1*\ 2.05) + (-1*2.57) + (1*0.73) \\ & = \mathbf{71.7/8} = \mathbf{8.9625} \end{aligned} \tag{5.8}$$

The values of different parameters using the values in Table 5.6 are shown in Table 5.7.

Table 5.7: Factorial Calculations

I	**A**	**B**	**C**	**Y**	**AB**	**AC**	**BC**	**ABC**
1	-1	-1	-1	49.77	1	1	1	-1
1	1	-1	-1	136.3	-1	-1	1	1
1	-1	1	-1	12.13	-1	1	-1	1
1	1	1	-1	26.49	1	-1	-1	-1
1	-1	-1	1	3.42	1	-1	-1	1
1	1	-1	1	2.05	-1	1	-1	-1
1	-1	1	1	2.57	-1	-1	1	-1
1	1	1	1	0.73	1	1	1	1
Total	97.68	-149.62	-215.92	0	-72.64	-104.1	145.28	71.7
Total/8	**12.21**	**-18.7025**	**-26.99**		**-9.08**	**-13.0125**	**18.16**	**8.9625**
SQUARES(q_i)^2	149.08	349.78	728.46		82.4	169.32	329.78	80.32

After the construction of Table 5.7, SST is calculated using equation 5.1.

$$SST = 2^3\left[(12.21)^2 + (-18.705)^2 + (-26.99)^2 + (-9.08)^2 + (-13.0125)^2 + (18.16)^2 + (8.9625)^2\right] \quad (5.9)$$

This gives,

$$SST = 15113.69015$$

Finally, the effect of different factors is calculated using:

$$(q_i)^2 * 8 \;/\; SST \qquad (5.10)$$

*(*Here *i denotes A, B, C, AB, BC or ABC)*. For example,

Effect of Algorithm selection from equation 5.10 = $(q_A)^2 * 8 \;/\; SST$

(where q_A= 149.08 * 8)

= 1192.64/15113.69015 = 7.89 %

Similarly, the final calculations in Table 5.7 based on SST values for each factor, are reported in Table 5.8. It can be concluded that the input size is most significant factor, which affects the performance of searching as it gives highest the SST score compared to other factors. This result confirms that input size and dataset factor play a vital role in evaluating the performance of the searching algorithms.

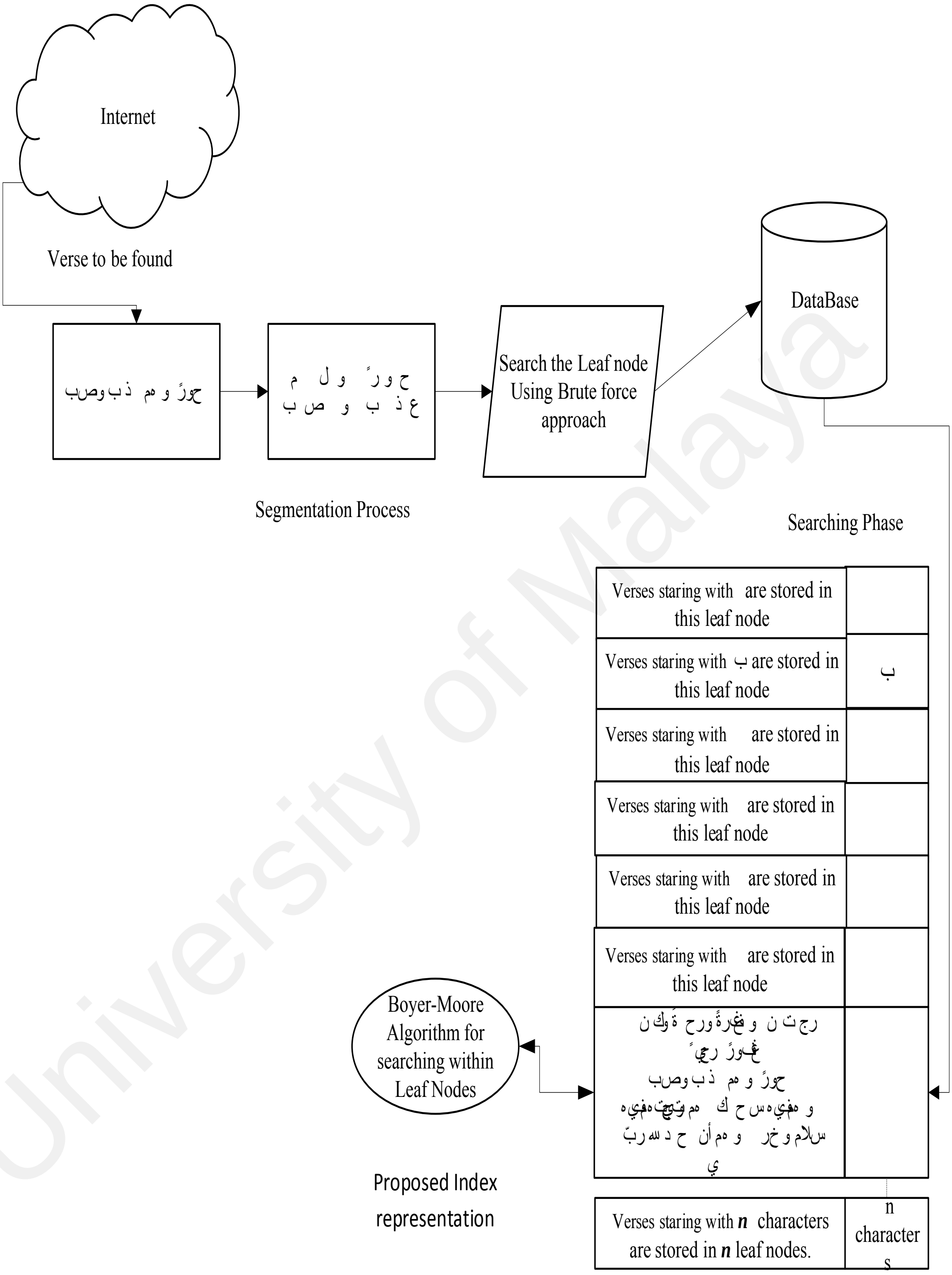
Table 5.8: Effect of different factors on retrieval

Effect of Algorithm selection	7.89%
Effect of Dataset	18.50%
Effect of Input size	38.50%
Effect of Algorithm and database	4.30%
Interaction b/w Algorithm and Pattern length	8.96%
Interaction b/w Pattern Length & Database	17.40%
Interaction b/w Algorithm, database, and Input	4.25%

In DHQ, since the input size cannot be changed, there is immense scope to arrange DHQ dataset in such way that can improve the retrieval process. The analysis of the literature review and results of factorial design motivated us to propose a new idea for representing data efficiently, such that optimal time efficiency is achieved, and the searching process improved.

5.3 Proposed methodology

To achieve efficient representation, we propose to segment the characters in the verses/sentences since each character is encoded by a unique code to organize the data. For segmentation, we explored the regular expression approach to remove the white spaces and segment the input verse into respective individual characters. After segmenting characters, the verse is extracted based on its first extracted character. This organization helps in retrieving the query word or verse quickly. An example of text retrieval from the database in the proposed framework is shown in Figure 5.1.



Internet
Verse to be found
حورً و مم ذ ب وصب
ح ور ً و ل م
ع ذ ب و ص ب
Search the Leaf node Using Brute force approach
DataBase
Segmentation Process
Searching Phase
Verses staring with are stored in this leaf node
Verses staring with ب are stored in this leaf node
ب
Verses staring with are stored in this leaf node
Verses staring with are stored in this leaf node
Verses staring with are stored in this leaf node
Verses staring with are stored in this leaf node
Boyer-Moore Algorithm for searching within Leaf Nodes
Proposed Index representation
Verses staring with n characters are stored in n leaf nodes.
n characters

Figure 5.1 : Proposed framework for searching vowelized diacritical texts

The steps involved in the proposed approach are listed below:

The complete Quranic texts involving Urdu, Uthmani and plain are sorted alphabetically. The verses are organized within the leaf nodes based on their first character and these leaf nodes are labelled accordingly based on the respective first character of verses contained within it. For Example, the leaf node containing the verses with characters “ ت” is labelled as “ ت”. This approach is repeated for all the respective verses.

The input verse to be searched is segmented into individual characters. The first segmented character is taken and mapped to leaf nodes. For example, if the first segmented character is “ت”, the whole search process is limited to a leaf node labelled as “ت”.

Finally, for searching the correct verse from a respective leaf node consisting of similar verses, we used Boyer Moore string matching algorithm.

The complete details are mentioned in below-mentioned sections:

5.3.1 Sorting Phase

In the sorting phase, the whole database containing diacritical text is sorted. Different characters are represented by different leaf nodes (*ln*). Each leaf node has a specific value based on the character that it represents. For example, a character “ ب“represents leaf node “ ب“. After creation of all leaf nodes, all verses are placed within their respective leaf nodes based on their first character. After sorting the database, the next step is character segmentation.

5.3.2 Character Segmentation

The aim of character segmentation in our proposed approach is to segment connected characters into individual characters. Most of the compilers and programming languages cannot process the connected Arabic verses accurately resulting in poor retrieval performance. There have been different encoding approaches proposed to represent individual characters. American Standard Code for Information Interchange (ASCII) is widely used as an encoding technique for English because each English character uses 7 bits with one extra bit to handle noise. Since ASCII uses 7 bits for character representation, it can handle 2^7 characters i.e. 128 English characters. (McEnery et al., 2000). However, this encoding scheme is inadequate to handle other non-Latin texts. Therefore, an 8-bit scheme has been proposed by the International Standard Organization (ISO) 8859 family (McEnery & Xiao, 2005). Further, to handle more complex characters, UNICODE 8, 16 and 32 encodings have been proposed (McEnery & Xiao, 2005). As a result, we use UNICODE-16 scheme for segmenting characters from Arabic verses, as it has variable length encoding and suits diacritical Arabic text and other complex texts. Sample Unicode representation for each Arabic character is shown in Table 5.9. Table 5.9 shows that each character has its own UNICODE representation.

This cue leads us to explore an approach using regular expressions to segment characters from Arabic text. A regular expression is a sequence of special characters that define the search pattern. There are many regular expression symbols that represent a particular operation. For example, "question mark (?)" indicates zero or one occurrence of the preceding element. Similarly, an "asterisk (*)" indicates zero or more occurrences of the preceding element. Similarly, in regular expressions, there are special operators known as **"**curly bracket denoted by {L}"**.** This expression maps each individual letter (L) to its unique Unicode number (Ilie, 2008). The steps are mentioned below:

Algorithm: Character Segmentation

Step 1: Calculate the length *l* of the input verse.

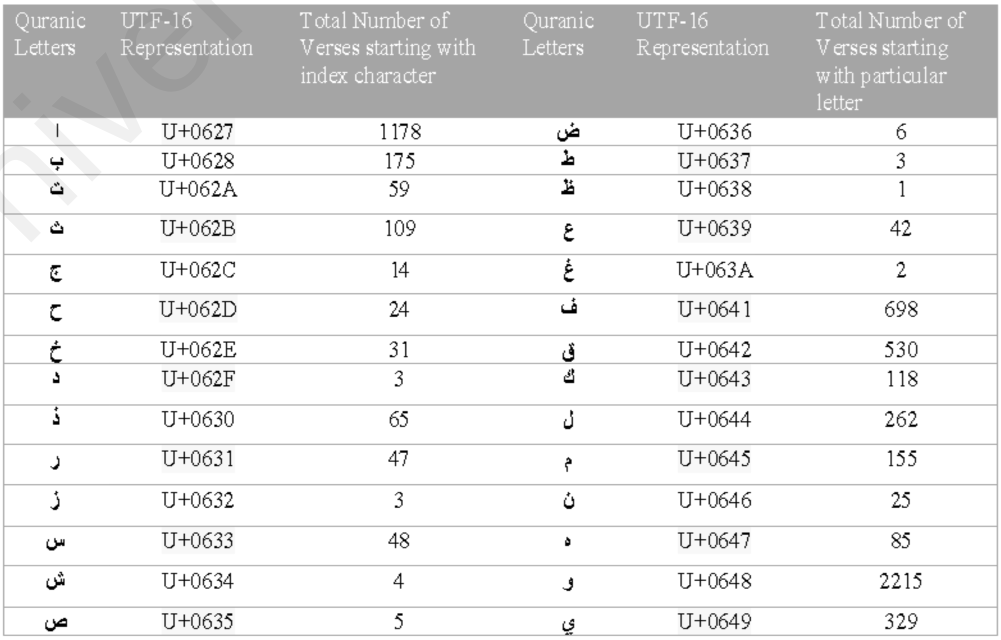
Step 2: Split the input string using a split method by specifying the start/end of each character using the regular expression delimiter.

Step 3: Using *for loop*, each character (specified by a delimiter) is iterated and mapped to its Unicode and the output is returned in the form of individual characters.

5.3.3 Proposed Data representation

As motivated by the arrangement of words in a dictionary, we propose to organize data based on segmented characters. In other words, the proposed method sorts text according to the first character in each verse (as shown in Table 5.9). We can see sample representations for digital Quran text, whereby the number of verses beginning with 'alif' (ا) is 1178, the number of verses beginning with 'baa' (ب) is 175 and so on. With this organization, we propose the framework shown in Figure 5.1 for representing diacritical sensitive texts as in the example of digital Quran. In this way, the proposed approach preserves the sensitivity of the Quran without losing information and retrieves the requested verses efficiently.

Table 5.9: Indexing approach for diacritical manuscripts



Quranic Letters	UTF-16 Representation	Total Number of Verses starting with index character	Quranic Letters	UTF-16 Representation	Total Number of Verses starting with particular letter
ا	U+0627	1178	ض	U+0636	6
ب	U+0628	175	ط	U+0637	3
ت	U+062A	59	ظ	U+0638	1
ث	U+062B	109	ع	U+0639	42
ج	U+062C	14	غ	U+063A	2
ح	U+062D	24	ف	U+0641	698
خ	U+062E	31	ق	U+0642	530
د	U+062F	3	ك	U+0643	118
ذ	U+0630	65	ل	U+0644	262
ر	U+0631	47	م	U+0645	155
ز	U+0632	3	ن	U+0646	25
س	U+0633	48	ه	U+0647	85
ش	U+0634	4	و	U+0648	2215
ص	U+0635	5	ي	U+0649	329

The main advantage of this data structure representation is that it can be extended to any language with minimal changes. For example, in the case of other related sensitive religious content, letters specific to that language can be used for indexing and the same procedure can be applied. There are 28 characters in Arabic which are also used for Urdu and Farsi texts. The steps for representation are as follows.

We divided the root node (r) into n leaf nodes (ln), where ln denotes the Arabic characters as shown in Figure 5.2 i.e. $ln =$ ا, ب up to ln^{th} Arabic character. Each leaf node (ln) has I identical children, i.e. a leaf node having a value of ln = "ب" with a Unicode value of U+0628 will have all I children starting with "ب" only. This organization helps not only in achieving time efficiency but also accuracy. Furthermore, this representation prevents hash collisions. Therefore, the search time is reduced from $O(m*n)$ in the worst case to $O(1) + O(m)$ *(m-denotes pattern to be found within I).* In this work, once the proposed approach identifies the leaf node, it uses the existing Boyer Moore (Boyer & Moore, 1977; Faro & Lecroq, 2013) algorithm for matching and retrieving data from the database.

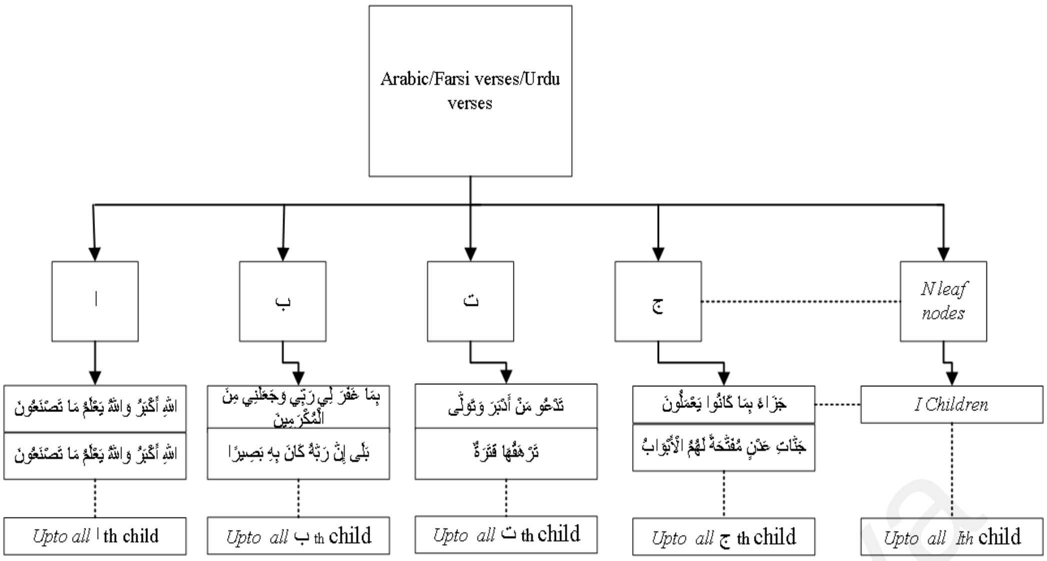

Arabic/Farsi verses/Urdu verses
ا
ب
ت
ج
N leaf nodes
اللهِ أَكْبَرُ واللهُ يَعْلَمُ مَا تَصْنَعُونَ
اللهِ أَكْبَرُ واللهُ يَعْلَمُ مَا تَصْنَعُونَ
بِمَا غَفَرَ لِي رَبِّي وَجَعَلَنِي مِنَ الْمُكْرَمِينَ
بَلَى إِنَّ رَبَّهُ كَانَ بِهِ بَصِيرًا
تَدْعُو مَنْ أَدْبَرَ وَتَوَلَّى
تَرْهَقُهَا قَتَرَةٌ
جَزَاءً بِمَا كَانُوا يَعْمَلُونَ
جَنَّاتِ عَدْنٍ مُفَتَّحَةً لَهُمُ الْأَبْوَابُ
I Children
Upto all ا th child
Upto all ب th child
Upto all ت th child
Upto all ج th child
Upto all Ith child

Figure 5.2: Proposed data structure design for diacritical Arabic and other related religious texts

The algorithmic steps and time complexity of the whole approach are as follows:

1. The Quranic database is sorted in a dictionary fashion shown in Table 5.9.
2. The input verse is segmented into individual characters using a regular expression approach.
3. The search and retrieval process are initiated by taking the first character of the input segmented verse and mapping that character with the leaf node.
4. Once a match is found, the remaining bits are matched with the children in that particular leaf node using the Boyer-Moore algorithm.

Time-complexity: Let *m* be the verse that needs to be searched from a given database of size *n.* The worst -case of time complexity is when *m* traverses through the whole database of size *n* and determines a match (if one exists) when all traversals are complete. The time complexity of our approach is constant i.e. *O(1),* since the algorithm does not need to traverse the whole database, but only traverses leaf nodes (whose size is fixed to 28 characters). However, when a particular leaf node is identified, then an extra time of

O(m) is needed to search within that leaf node. Hence, the total time complexity is *O(I+m)*.

5.4 Results

To evaluate the performance of the proposed method, we use some well-known databases, including Arabic data-sets, Uthmanic data-sets and Urdu datasets from tanzil.net (tanzil.net, 2016) and hamariweb (Hamariweb, 2018) respectively. The sizes of those data sets are 1.24, 1.30 and 2 MB, respectively. The performance of the proposed method is measured by calculating the segmentation accuracy for the segmentation steps and search time for the data representation steps. To show the effectiveness of the proposed approach, we compare our algorithm with the existing methods that includes; Quran Quote Verification (QVT) algorithm (Alshareef & Saddik, 2012) and the traditional MySQL approach that uses the linear search algorithm(Greenspan & Bulger, 2001). All of these existing approaches have been explained in chapter 2 respectively. In our comparative study, the implementation of the other algorithms was achieved using NetBeans 8.02 on i-5 Intel Processor with 4 MB cache, 4 GB RAM using Windows 10.

5.4.1 Experiments for tokenization

Sample qualitative results of the proposed segmentation approach for segmenting Arabic and Urdu characters from text lines are shown in Table 5.10, where one can observe that the proposed segmentation had worked well for the three types of data. The reason for the efficiency of our segmentation can be attributed to the fact that the mapping of individual/partial Arabic characters to its respective Unicode representation, and extracting full characters from that Unicode had allowed for efficient matching.

Table 5.10: Tokenization of simple partial diacritical Arabic texts

Different Non-Latin Texts	Sample Verses	Processing using QVT Algorithm	Processing Using Proposed Approach
Plain Arabic	تل أوح يك لتك ب وقِم صلاة ن صلاة تنه فح و نكر و لكر لله أكر و لله م تصن ون	تل أوح يك كت ب وقِم صلاة ن صلاقتن ه فح و نكر و لكر الله أك ر والله يعلم تصرن ون	ت ل م أ و ح ي ل ي ك م ن ل ك ت ب و أ ق م لصّ ل ة نّ لصّ ل ة ت ن ى ع ن ل ف ح ش و ل م ن ك ر و ل ذ ك ر للّ أ ك ب ر و ل لّ ي ع ل م م ت ص ن ع و ن
Urdu	ق لکے رے یرکي کیں ہم وگ	ق لکے رے یرکيکیں ہم وگ	ق ب ل ک ے ب ر ے م ی ں ک ی ک ی ں م ل و گ
Uthmanic Arabic	ك ن ر د حرث ٱل خرة ز فُ حرثۦ و ك ن ر د حرث ٱ لدُّي ؤتۦ ن ه و فُ ٱل خرة صيب	ك ن ر د حرث ٱل خرة ز ف حرث و ك ن ر د حرث ٱ لدي ؤت ن ه و ف ٱل خرة صيب	م ن ك ن ي ر ي ح ر ث ٱ ل خ ر ة ن ز ل ُ ف ى ح ر ث ۦ و م ن ك ن ي ر ي ح ر ث ٱ ل ّ ن ي ن ؤ ت ۦ م ن و م ل ُ ف ى ٱ ل خ ر ة م ن نّ ص ي ب

In Table 5.10, it is observed that the QVT algorithm is not able to segment the Urdu texts since the QVT algorithm is developed only for diacritic texts. The segmentation in QVT is based on suffix and prefix rules, rather than tokenizing verses into individual characters. On the other hand, the proposed approach segments each Urdu verse into separate characters properly. To validate the strength of the proposed segmentation step, we conduct experiments on Uthmanic Arabic verses, which use more diacritics compared to plain Arabic verses. When we compare the segmentation results of QVT method, the proposed approach is better than the QVT, since QVT fails to correctly segment the text (Table 5.10).

Quantitative results of the proposed and existing methods are reported in Table 5.11, where the segmentation accuracy of our proposed segmentation is higher than the existing methods. The reason for poor segmentation accuracy of the existing method is that the method is developed for Latin and non-diacritic text but not for non-Latin text with diacritics.

Table 5.11: Accuracy of text segmentation for all three data sets

Text Types	Sample Verses	QVT	Proposed Approach
Plain Arabic text	تل أوح يك لكتب وأقم صلاة ن صلاتنه فح و نكر و لكر لله أكر و لله م تصن ون	25.47%	98.5 %
Urdu texts	قل كـے ر ے يي كيكـيں ہم وگ	46 %	100 %
Uthmanic Arabic text	كن ر د حرث ٱل خرة ز فُ حرث و كن ر د حرث ٱ ثّي ؤ نه و فُ ٱل خرة صيب	29.3%	95.2 %

5.4.2 Experiments for Data Representation

Quantitative results of the proposed and existing methods are reported in Table 5.12 where it is noticed that the proposed method achieved the best time efficiency compared to the existing methods. To calculate the time for retrieval, we choose random verses as queries for searching the respective databases. Finally, the average time for random verses is reported in Table 5.12. The time efficiency is calculated as:

$$Time\ efficiency = \frac{P.A\ response\ time - E.A\ response\ time}{E.A\ response\ time * 100} \qquad (5.11)$$

Here, *P.A* denotes the proposed approach and *E.A* denotes existing approaches. It is noted from Table 5.12 and equation 5.11, that the proposed approach had achieved 61 %, 72%, 84%, and 62% improved time efficiency as compared with the B+ tree approach, Muslim-Web, Search-Truth and the QVT approach, respectively. The main reason for poor accuracy is due to the presence of different diacritics that increases the time complexity of an algorithm and existing data-structure approach, where all those verses are being stored in a serial fashion. This serial organisation of data within the database is another factor for the poor performance of the existing approaches. From the results, it can be concluded that the proposed segmentation and representation combination has the ability to achieve efficient time complexity, irrespective of the database size, and can be extended to other languages with minimal changes.

There are numerous religious scriptures in Chinese and Sanskrit that involve diacritics for reading purposes. Advantageously, our approach can be applied or extended for other diacritic based languages. Instead of using Arabic characters for single verse extraction, the characters of Sanskrit or Chinese can be used to extract the given verse based on its first character. The methodology will be the same, only the input will be different.

Table 5.12: Search time for different Diacritical Texts (in milliseconds)

Different Text samples	B+ Tree Index Method (MySQL)	Muslim-Web engine (Binary Search Algorithm)	Search-Truth (Linear Search Algorithm)	QVT Algorithm	Proposed Approach
تل أوح يك لكت ب وأُم صلاة ن صلاقتنه فح و نكر ولكر لله أك ر و لله م تمن ون	964.2	2614	2741.25	980.2	380.2
قالک ےار یجہم کیا کیں ہملوگ	321.2	-	-	-	240.4
ك ن ر د حرث ٱل خرة ز فُ حرثۓ و ك ن ر د حرث ٱلتُّي فۓ نه و فُ ٱل خرة صيب	912.4	No result	No result	928.4	370.1

5.5 Summary

In this study, we propose a new approach introducing a novel method for segmenting characters from verses and representing data using segmented characters to efficiently retrieve diacritical texts like digital Quranic text in plain and Uthmani script as well as

Urdu and Farsi. This study explores the regular expression approach for segmenting characters from verses inspired by the Unicode-16 encoding scheme. The proposed approach involves a novel indexing method using segmented characters to represent the data in such a way that optimal time efficiency can be achieved, irrespective of database size and language. Experimental results for the segmentation process show that the proposed segmentation approach outperforms the existing methods in terms of segmentation accuracy. Similarly, the experimental results on validating the indexing and data representation approach show that the there is a significant improvement in the search time. This constitutes the first attempt to develop a segmentation-based data representation approach for retrieving diacritical texts such as Arabic, Urdu, and Farsi texts.

As part of future work, we plan to extend this approach for other scripts, such as Chinese and Sanskrit and develop an expert prototype system for real-time applications. The limitation of this approach lies in the inability to extract two verses simultaneously. To overcome the issue of authenticating more than one verse at a time, we propose a second method based on string or pattern matching algorithms as discussed in Chapter 6.

CHAPTER 6: NEW SPLIT BASED SEARCHING METHOD FOR EXACT PATTERN MATCHING

6.1 Introduction

The pre-processing phase and authentication phase involving single verse has already been discussed in Chapters 4 and 5 respectively. This chapter presents the third method as part of our research, its purpose being to overcome the limitation inherent in Chapter 5. The method proposed in the previous chapter is suitable for identifying a single sentence or Quranic verse only. To authenticate (search and retrieve) more than one verse or sentence at a time, we have explored pattern or string matching algorithms. The proposed approach is based on exact matching algorithms. The existing exact matching algorithms have been found to be unsuitable for Arabic texts like Quranic texts. Hence, a new algorithm is proposed by way of splitting the verses into smaller units. Two different algorithms are proposed to shorten the search time and improve memory consumption for authenticating multiple verses. Firstly, the concept of exact or pattern-based approaches and related research is discussed in Section 6.2. The first approach algorithm (A1) is discussed in Section 6.3, and the second approach algorithm (A2), a modified version of the first approach (A1) is discussed in Section 6.4. A performance evaluation of both approaches is finally presented in Section 6.5. The chapter is concluded in section 6.6.

6.2 String/Pattern Matching Approaches

6.2.1 Motivation

As swift changes occur in digital technologies, the conversion of raw data to digital data online is also changing with the same proportionality. As a result, the size of the database increases drastically. Therefore, to cope with real-time applications and situations, there is a need for focussing on both time and space complexity of the systems or methods because those two parameters decide the usefulness and effectiveness of the system, despite that methods achieve good accuracy. Most of the existing methods in the literature have focused on time complexity, and little attention has been paid to the issue of space complexity (memory consumption). Therefore, there is a need for developing a method which achieves both time as well as space efficiency, irrespective of the size of the database (Frakes & Baeza-Yates, 1992).

It is evident that in recent days, modern programming languages, such as Java and C# are widely used for setting up real-time systems because those software languages involve automatic memory management (Yang et al., 2004). It is noted that heap size, which is part of memory segment plays a major impact on the performance of garbage collection, which in turn affects the overall performance of the systems with multiple processes (Kim & Hsu, 2000). For example, if the heap size is less than the application requirement, it would cause an excessive garbage collection, while a heap size greater than the physical memory results with induced paging.

On the other hand, there is no generalized criterion to decide the correct heap size according to application requirements (Yang et al., 2004). This is beyond the scope of this work. One such illustration using existing string matching (Al-Ssulami, 2014) on Arabic datasets is shown in Figure. 6.1, where we can see that the algorithm initially requested 350 MB of heap, but uses 70 MB (on average), resulting in a wastage of memory resources. Therefore, it is necessary to focus on both time and space complexities of the method.

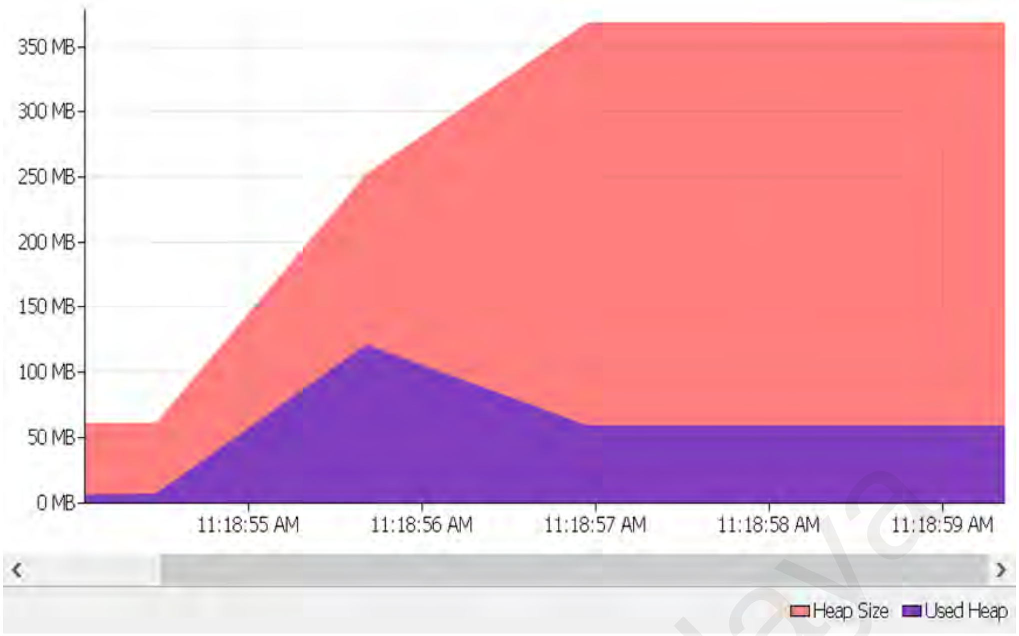


Figure 6.1: Memory Usage of Existing Exact Matching Algorithms.

The main reason that existing exact string matching algorithms consume more memory is that pre-processing is involved in the computation of shifts. For example, in Figure 6.2, the Boyer-Moore algorithm (Alfred, 2014; Boyer & Moore, 1977; Lin et al., 2013; Rahman et al., 2017) starts searching characters from right to left of the given query pattern. If there is a mismatch, algorithms shift as many as *m* characters according to the shift table computed in the pre-processing phase. It looks similar to the Quick search (QS) algorithm (Sunday, 1990) with respect of finding matches, except that the BM algorithm uses both good suffix shifts and bad-shifts, while the QS algorithm uses only bad shifts (Lin et al., 2014).

BM is one of the most standard and widely used algorithms in pattern matching. Many improvements in terms of time efficiency were carried out by researchers studying the concept of character shifts (Al-Dabbagh et al., 2017; Jaiswal, 2014; Nsira et al., 2015; Saleh et al., 2015). Few existing string matching algorithms using the same concept include: fast search searching algorithm (Lecroq, 2007), modified Boyer Moore algorithm (Faro & Lecroq, 2013; Rafiq et al., 2004), Horspool algorithm (Horspool, 1980a), Tuned

BM (Sunday, 1990), Turbo BM (Crochemore, 1994), SSM Algorithm (Al-Ssulami, 2014) and so on.

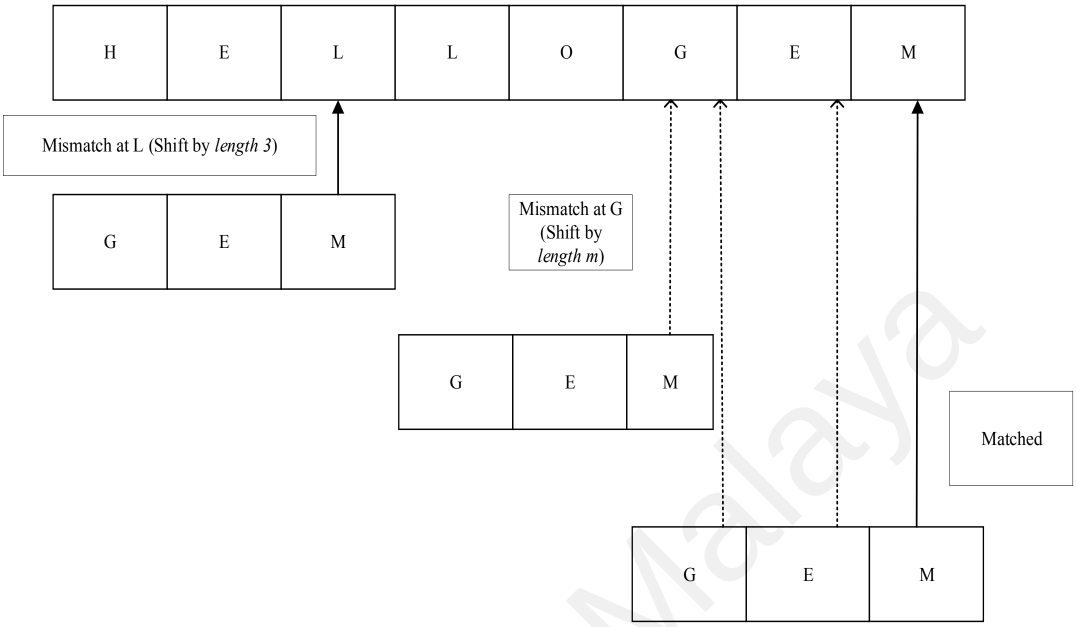

H
E
L
L
O
G
E
M
Mismatch at L (Shift by *length 3*)
G
E
M
Mismatch at G (Shift by *length m*)
G
E
M
Matched
G
E
M

Figure 6.2: Boyer Moore Algorithm (Lin et al., 2014).

Several algorithms were proposed to overcome the drawback of the above-mentioned process based on characters. For example, (Horspool, 1980a) simplifies the Boyer-Moore's algorithm by removing the good-suffix rule (Boyer-Moore-Smith Algorithm). (Michailidis & Margaritis, 2002) proposed algorithms which are extensions of the BM algorithm, focusing on computing the shift with the text character. Timo Raita (Raita, 1992) proposed an algorithm known as the Raita algorithm, which is a modified form of the BM algorithm. (Crochemore, 1994) proposed a Turbo-BM algorithm, which works based on a dynamic simulation technique. Berry-Ravindran (Berry, 2001) proposed an algorithm, known as Berry and Ravindran algorithm, which is an improvement over the quick-search algorithm. Ahmad (Ahmad, 2014) proposed an idea of exploring parallel processing for the two pointers that are used in the string matching process, i. e., one pointer starts searching from the left side and another pointer starts searching from the right side, thus reduces the overall search time. (Karp & Rabin, 1987) proposed a hashing technique to avoid a quadratic number of character comparisons (Lecroq, 2007). However, the drawback of this approach is the possibility of hash collisions. Similarly, there are bit-parallelism, automata-based, and bit-wise exact matching approaches to improve the search time.

In light of the above discussion, it can be asserted that the primary focus of the existing method is time complexity (Bobroff et al., 2016), (Shaham et al., 2001). Previously, many researchers ignored the fact that space complexity (memory consumption) is also one of the major factors to achieve time efficiency, particularly when the database size increases continuously.

Thus, in this chapter, we present novel approaches to solve the exact string matching problem, which achieves both time and space efficiency for natural language texts, specifically Arabic texts. All those texts apply different encoding schemes, such as the ASCII-based encoding scheme, the Unicode Transformation Format (Nuaymi) and so on (McEnery & Xiao, 2005).

Elaborate Arabic script patterns are complex to search and retrieve, compared to ASCII-based text (Alfaifi & Atwell, 2016; Elayeb & Bounhas, 2016; Metwally et al., 2016). Searching and retrieving diacritical Arabic patterns are more complex than simple Arabic patterns (Metwally et al., 2016). The presence of symbols, diacritical signs, elongated characters and other such elements increase the complexity of searching Arabic texts and also the time complexity (Abdelali et al., 2004; Darwish & Magdy, 2014). Digital Quranic text, which is written in Arabic, constitutes a very suitable example, given its highly complex diacritical texts (Al Gharaibeh et al., 2011; Alginahi et al., 2013; Farghaly & Shaalan, 2009; Khalaf et al., 2014). These diacritical texts contain a large number of additional symbols that decrease the retrieval process and increase the search time. Although the search time can be improved by removing all symbols and diacritics, this will obscure the meaning of the Quranic verses. Arabic Quran verses are very sensitive to the arrangement of diacritics. The removal of a single diacritic or symbol can change the meaning of the whole verse (Alshareef & Saddik, 2012; Arslan, 2015; Ibrahim, 2010; Mohammed et al., 2015) as explained in previous chapters. Thus, it is imperative to search for a diacritical pattern without removing any of the symbols or diacritics.

6.2.2 Classification of String/Pattern matching approaches

The methods for pattern matching can be classified into two categories: single pattern matching and multi-pattern matching (Faro & Lecroq, 2013). The single pattern matching approach consists of identifying a single pattern from the whole database. The multi-pattern approach consists of identifying multiple patterns from a single database (Alhendawi & Baharudin, 2013; Hudaib, 2008). This study focuses exclusively on single pattern matching as multi-pattern matching extends beyond its scope.

The workable methods based on single pattern matching approaches can be divided into four categories as mentioned above: character-based, hashing, bit-parallelism, and automata-based approaches (Yuan et al., 2010). The character-based approach searches for the pattern at the character level. This approach includes the two key stages of searching and shifting. Considering all the existing character-based approaches, the baseline approach consists of the Boyer Moore algorithm (Boyer & Moore, 1977). Since character-based approaches work at the character level, the methods are computationally expensive. Hash-based approaches find hash values for the characters to match rather than the characters themselves. Automata-based approaches involve automata theory for finding states as a suffix for matching. Such methods generally achieve better results, however, consume more memory due to the necessary state diagram construction and traversal. Finally, bit parallelism approaches involve parallel processing to speed up the matching process (Faro & Külekci, 2013; Faro & Lecroq, 2013). The main issue with bit parallel approaches is the dependence on the computer word-size for matching, and the difficulty in implementation (Hennessy et al., 1999; Lecroq, 2007). Bit parallel algorithms are harder to write than sequential algorithms as bit-parallel based algorithms normally use concurrency that results in potential software bugs (Hennessy et al., 1999).

In general, the character-based approaches are simpler and easier to implement than other approaches due to ease of manipulating shifts.

6.2.3 Related Work

The BM algorithm (Boyer & Moore, 1977) is considered as one of the standard benchmarks in the area of string/pattern matching literature (Hume & Sunday, 1991). In the BM algorithm, searching starts from right to left for any given text. The scanning stops once a complete match has been found. However, in the case of a mismatch, it uses a shifting process commonly known as 'bad shift' and 'good shift' (Ahmad, 2014). Those shifting tables require significant pre-processing in terms of calculations; such as how many characters need to be skipped if there is a mismatch; how many characters need to be skipped if there is a match, and so on. Many attempts have been made to improve BM algorithms in terms of their required search time and include the Horspool algorithm (Horspool, 1980a), the Tuned BM algorithm (Sunday, 1990), and the Turbo BM algorithm (Crochemore, 1994). The Turbo-BM algorithm is based on the dynamic simulation technique. Since our aim is to achieve time efficiency for searching Arabic text from a database, we review the time complexity of the existing methods in this section.

In this study, the worst-case scenario is referred to as the Big O notation. Let *p* be the pattern to be searched with length *m*, and *t* the source text of length *n*. Those are the symbols used for reviewing the time complexity of the existing methods as reported in Table 6.1.

- *Boyer-Moore algorithm:*

  It needs two rules for identifying and locating patterns, namely a good suffix rule, and a bad character rule. Let the necessary shift to be used in case of a mismatch be denoted by s. Thus, in case of a mismatch, the good shift rule aligns the pattern p of length m over text t in such a way that t [s+i+1…….s+m-1] = p [i+1…m-1]. To calculate the number of shifts, the BM algorithm requires pre-processing that occupies O (m) of space, and in the worst-case scenario requires O (mn) of time. Hence, the overall complexity of BM becomes O (m)[n+1].

- *Tuned Boyer-Moore (BMT) algorithm:*

  This algorithm consists of the two phases: last character localisation and matching. The first phase consists of the searching pattern (p (m-1)) using three rounds of blind shifts based on the bad character rule of the BM algorithm. In the matching phase, pattern p (0…m-2) is tested to obtain a match with the corresponding characters of the given text, t. This algorithm has a quadratic time complexity denoted as O (n^2) in the worst-case scenario and can increase further by trying to find a match between the pattern and a given text during n blind shifts. Like the BM algorithm, BMT also requires O (m) of space, thus has an overall complexity of O (n^x+m), where x denotes an exponential power.

- *SSM algorithm:*

This algorithm represents a modification of the Horspool (Horspool, 1980a) algorithm. In fact, it is a modification of the BM algorithm where the good suffix rule is dropped. In the work by Faro (Faro & Lecroq, 2013), Hash 3 and simple backward non-deterministic matching algorithm (SBNDM algorithms) have shown better results among 85 algorithms for natural texts like the bible. For that reason, the SSM algorithm (Al-Ssulami, 2014) has been compared with Hash 3 and SBNDM algorithms. The SSM algorithm has shown better results than Hash 3 and SBNDM, indicating that the results are better than those existing 85 algorithms. The increments in the shifts are based solely on the bad character rule. The algorithm works by scanning text from the leftmost end and matching patterns from an opposite side. The algorithm searches the pattern using the maximal shift approach, where the shift is calculated based on the position of a pivot character. In the case of a mismatch, the Horspools shift (where initially a table is constructed to determine character shift) is applied. This algorithm again involves pre-processing for computing all those shifts and takes O (m) space with sub-linear time complexity. Thus, the overall time complexity is O (m)[n+1].

- *Brute Force Algorithm:*

This is one of the simplest algorithms that does not involve any pre-processing. It searches the whole pattern *p* from a text *t*, character by character in a serial manner until the whole pattern is found. The only limitation of this approach is its time complexity. The overall time complexity of this algorithm is linear, i.e. *O (mn)*. However, the same algorithm can perform faster on more advanced machines where the instruction set, and processing speed is massively increased. Our proposed approaches are based on the same assumption that brute force algorithms can perform better on modern machines for UTF based texts like Arabic. The summary of the above-mentioned algorithms is shown in Table 6.1.

Table 6.1: Summary of Existing String based approaches (m denotes pattern length and n denotes the length of a given text)

Algorithms	Time complexity	Limitations	Dataset used
BM	Pre-processing: *O (m) of space* Searching Phase: *O (m*n)*	Not suitable for small patterns.	English texts
BMT	Pre-processing: *O (m)* of space Searching Phase: O (n^2)	This is good for the short pattern but not long pattern like sentences (Charras, 2004).	UNIX dictionary
SSM	Pre-processing: *O (m)* of space Searching Phase: O (*n+1*)	Performance is not constant. Search time varies for each pattern length and dataset.	Natural texts, protein, genome
Brute force	Searching Phase: *O (m*n)*	Not suitable for large datasets due to the character by character matching process.	Natural texts

Let t_L be the lookup time taken by the above mentioned exact matching algorithms for shifting a pattern *p* from a given text *t*. For $_{mismatch}$ mismatches, the existing exact matching algorithms need $n(t_L)$ of time to decide when to shift and when not to shift. This changes the overall time complexity of the above-mentioned algorithms to:

$$O(m)\ [n_{mismatch}+1] + t_L \qquad (6.1)$$

where t_L can be negligible in case *p* is found in the first attempt. However, in the worst case, it will be $n_{\max}$ times. This factor of t_L that arises due to pre-processing can be avoided following the brute force approach, as it does not involve any pre-processing. Although all those algorithms have competitive time complexities, the performance of those algorithms differs based on the different encoding techniques as shown previously in Chapter 5, Table 5.4. From the observations of Table 5.4, it showed that the existing string matching algorithm performance degrades for UTF based texts like Arabic. This is due to the extra symbols and characters used in those texts that occupy extra bits, thus increasing the search time. However, the brute force approach performed better for both the texts. This factor motivated us to propose an algorithm that uses a character-based approach of the query pattern for searching in the database.

From the above discussion, it is observed that the existing methods mostly use natural texts like English for searching. The same approach may not be useful for searching Arabic text from the databases. Thus, we propose two methods for searching Arabic text, which consumes less time and memory compared to the already existing methods.

6.3 Split-Right based search approach-A1

As noted from the literature review, the conventional exact string-matching algorithms search query patterns at the character level, resulting in more processing time and more space. This factor motivated us to propose an algorithm that uses a number of characters from the query pattern for searching in the database. Here, the proposed algorithm splits the query pattern, say, *p*, into two equal halves, say *p1*, *p2*. If the string pattern length is even, it considers *p2* to find a match with the dataset. Once algorithm finds a match, it matches *p1* with the adjacent characters found for *p2* directly. As a result, it reduces the number of comparisons and reduces memory consumption. The steps of the proposed algorithm are shown in Figure 6.3.

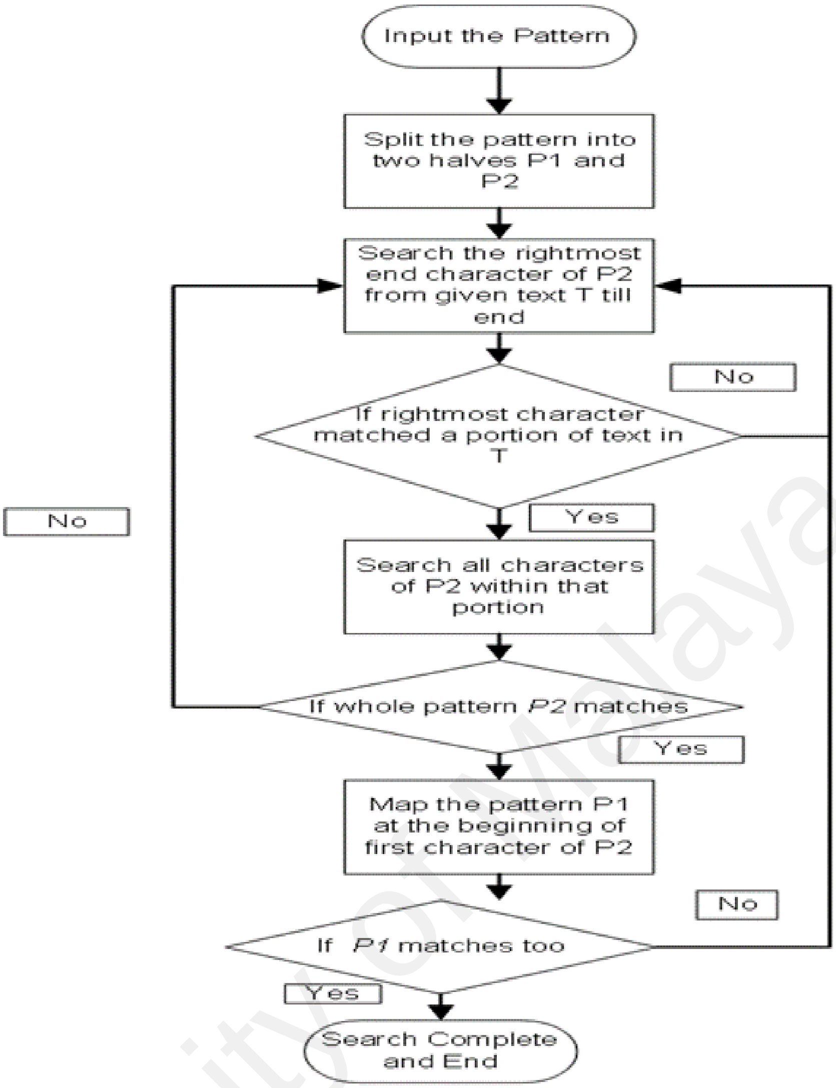

Input the Pattern
Split the pattern into two halves P1 and P2
Search the rightmost end character of P2 from given text T till end
No
If rightmost character matched a portion of text in T
No
Yes
Search all characters of P2 within that portion
If whole pattern P2 matches
Yes
Map the pattern P1 at the beginning of first character of P2
No
If P1 matches too
Yes
Search Complete and End

Figure 6.3: Flowchart of the proposed algorithm.

The algorithm is now described as follows. Suppose we need to find a pattern *p* of length *n* from a given text *t*. The proposed algorithm searches p_2 with length m_2 only, such that ($m_2 <= length\ of\ a\ pattern\ p$) using a brute force algorithm (i.e. each character is checked character by character). Once the right half has found a match during the search process, the algorithm considers the whole left half string to match within the database by considering a reference found by the right half string match. As a result, the proposed algorithm involves shift only operations for the right half, in contrast to the brute force algorithm or traditional exact matching algorithms, which involve many shifts for a whole pattern. The proposed algorithm starts scanning from the rightmost end to the leftmost end of the given text t, and the matching process of p_2 starts from left to right i.e. ($i_0, i_1 \ldots i_{nth}$-1). Here, i denotes the index of the given text where p_2 found a match. In case there is a match, p_1 is mapped onto the location using the equation below, where i_0 denotes the position of i_{th} character of t, where the rightmost character of p_2 matched.

$$Position\ to\ Map\ (p_{map}) = \mathrm{t}[i_0] - m_2) \tag{6.2}$$

From equation (2), if p_1 also matches the given text, the algorithm moves to another location to verify the other matches. In case there is a mismatch, the algorithm again starts scanning from the last matching position, i.e. i_0.

The proposed algorithm is illustrated in Figure 6.4 for a query string of even lengths. Figure 6.4 shows that for a given text (*t*): “HELOGEMLED”, the proposed algorithms divide a query string “OGEM” into “OG”, (say *p1*) and “EM” (say *p2*). Since the query string length is even, it divides it into two equal sub-strings. Therefore, the proposed algorithm scans *p2* in *t* until it finds matches for “*M*”. The rightmost character “*M*” is denoted by r_0. In the first search phase, “*M*” does not match with “*E*” in a given text (*t*). Thus, the pointer will move to the r_0+1 position, again encountering a mismatch with “*L*”, followed by “*O*”, “*G*” and “*E*”, respectively. Next, there is a match where character “*M*” of sub-pattern (*p2*) matches character “*M*” in the text (*t*) at location i_o. Now, the pointer moves to a position (i_0-1) in the given text. Finally, the next character of the pattern i.e. “*E*” also matches the text, indicating a match. At last, *p1* is mapped directly based on the location of the last match.

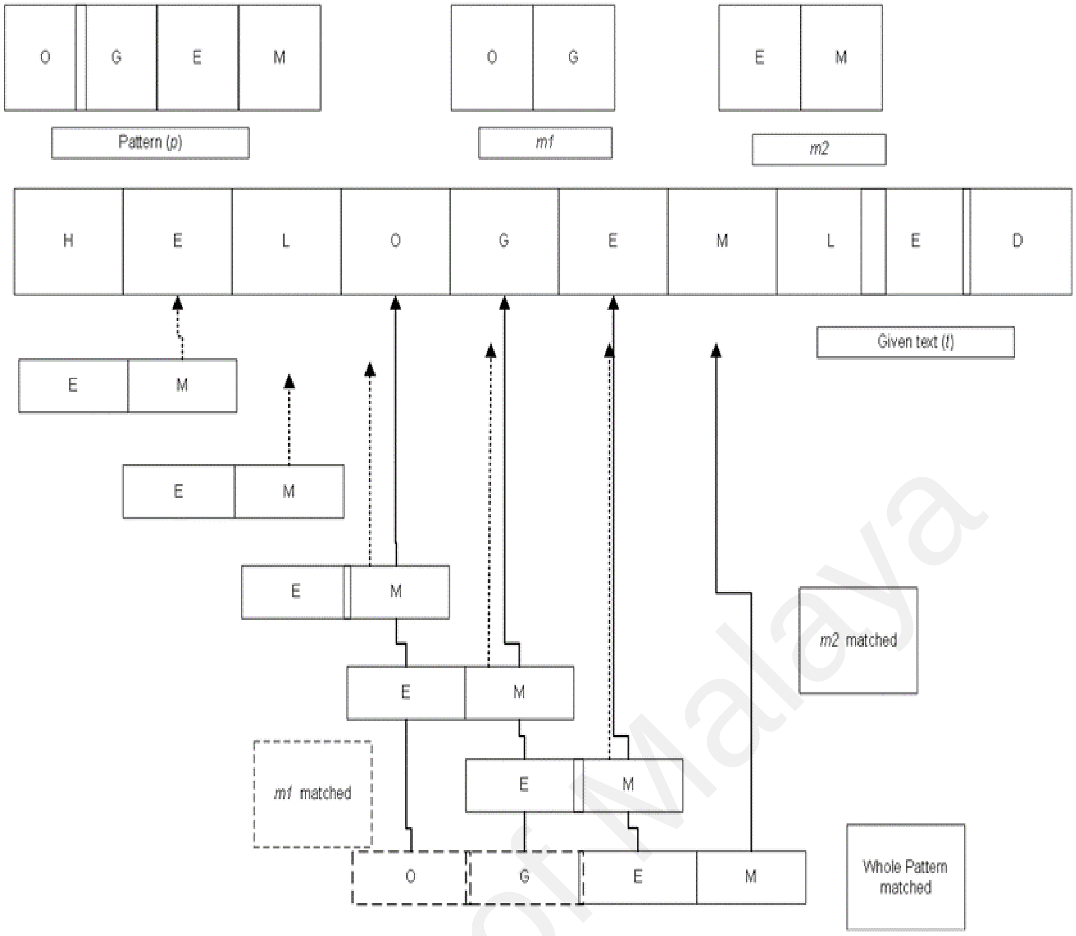


Figure 6.4: Working example of a proposed algorithm for even length pattern (dotted lines represent ongoing search process and solid lines indicate the final match found)

It must be noted that the proposed algorithm considers a single string as a query word. In other words, the query does not have multiple independent words. As a result, once the proposed algorithm finds matches in the right half, it is obvious that the left half should be associated with the right half. Therefore, there is a very little probability of left half being located in one place in a database and the right half at another location. According to the logic, if *p2* is matched, there is no need to search for *p1*, as the probability of *p1* having a match is also likely to be high, as per the probability equations from (Feller, 1968):

If *p1* matches 50 % of the pattern (pattern match success is given by:

$$P_{match\,success\,rate} = \frac{p2}{(p1 + p2)} * 100 \qquad (6.3)$$

= 50 % (for even string)

However, in terms of accuracy-related issues, we also match *p1* to make sure the pattern is accurate. The time to process *p1* will be negligible due to the concatenation of *p1* with *p2.*

6.4 Parallel-Split search-based approach-A2

In the second approach (A2), the aim was to improve the searching time process for natural language texts and especially Arabic texts in particular. The whole input pattern *p* is split into individual character components as shown in Figure 6.5. The reason for splitting the pattern is to ease the search process involving diacritical texts. The proposed approach estimates the total length of the input pattern and divides the input pattern into two sub-patterns based on the midpoint of the input pattern (as in approach A1). The output of the tokenisation is passed to the search phase. In the search phase, the two sub-patterns, namely the *left half* and *right half,* are processed simultaneously compared to Approach A (where only the right half was processed). Both those halves (i.e. *left* and *right*) search their specific patterns simultaneously from the given text. As a result, it reduces the number of comparisons needed and reduces the memory consumption compared to existing pattern matching algorithms. The reason for consuming less memory compared to existing character-based approaches is that the proposed method does not require pre-processing; such as information about shifting when a correct match is found or when a mismatch is found during searching. This pre-processing leads to high memory consumption and requires more time, as it uses a lookup approach for the mismatch.

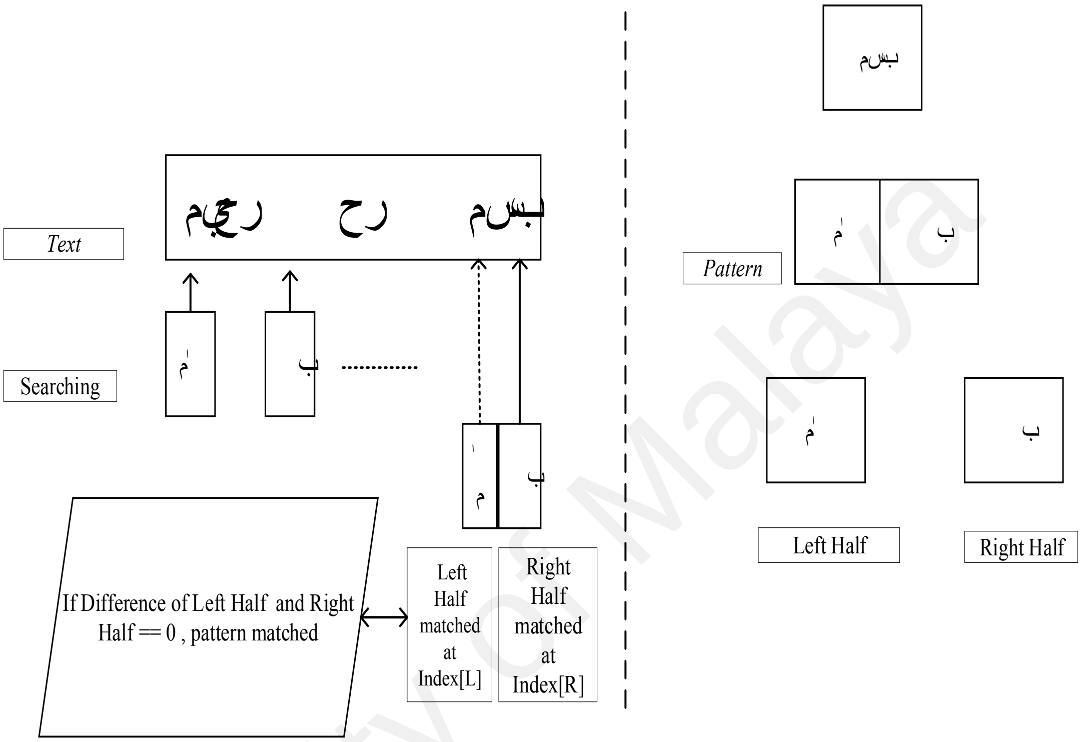

Text
Searching
Pattern
بسم
مٰ
ب
....
Left Half
Right Half
If Difference of Left Half and Right Half == 0 , pattern matched
Left Half matched at Index[L]
Right Half matched at Index[R]

Figure 6.5: Partition based approach to search Arabic diacritical texts

The steps of the proposed approach are illustrated in Figures 6.6 and 6.7, respectively. In Figure 6.6, the given text is “SAQSAQI” and the search pattern required is “SAQI”. According to the proposed approach, the whole input pattern will be split into two sub-patterns: the left half having the pattern “SA”, and the right half having the pattern “QI”. Following the splitting process, both patterns are searched simultaneously. In Figure 6.6, the left half is found at index (1) and right the half is found at index (Jensen et al.). In order to find the exact pattern, both the halves must have a difference of zero between each other. However, in the present scenario, it can be seen that the difference between the two halves is greater than zero, indicating a mismatch.

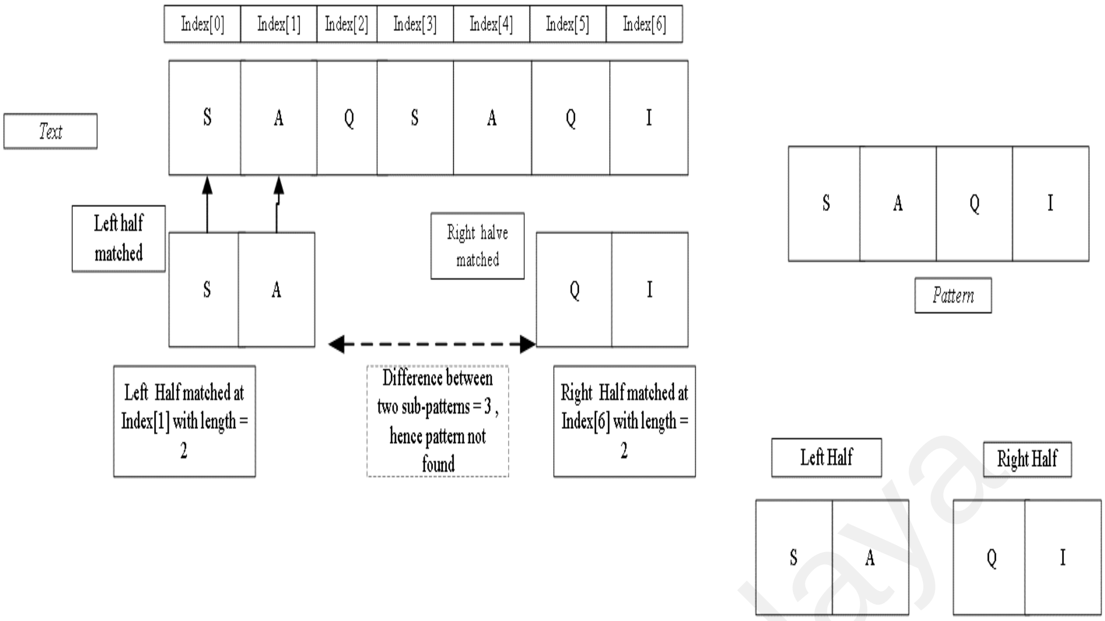

Index[0]
Index[1]
Index[2]
Index[3]
Index[4]
Index[5]
Index[6]
Text
S
A
Q
S
A
Q
I
Left half matched
Right halve matched
S
A
Q
I
Left Half matched at Index[1] with length = 2
Difference between two sub-patterns = 3 , hence pattern not found
Right Half matched at Index[6] with length = 2
S
A
Q
I
Pattern
Left Half
Right Half
S
A
Q
I

Figure 6.6: Example of an unsuccessful pattern search

In the first scenario, there was no match. The same procedure is again repeated as shown in Figure 6.7. However, in this case, there is no difference between the left half and the right half, thus indicating a complete match.

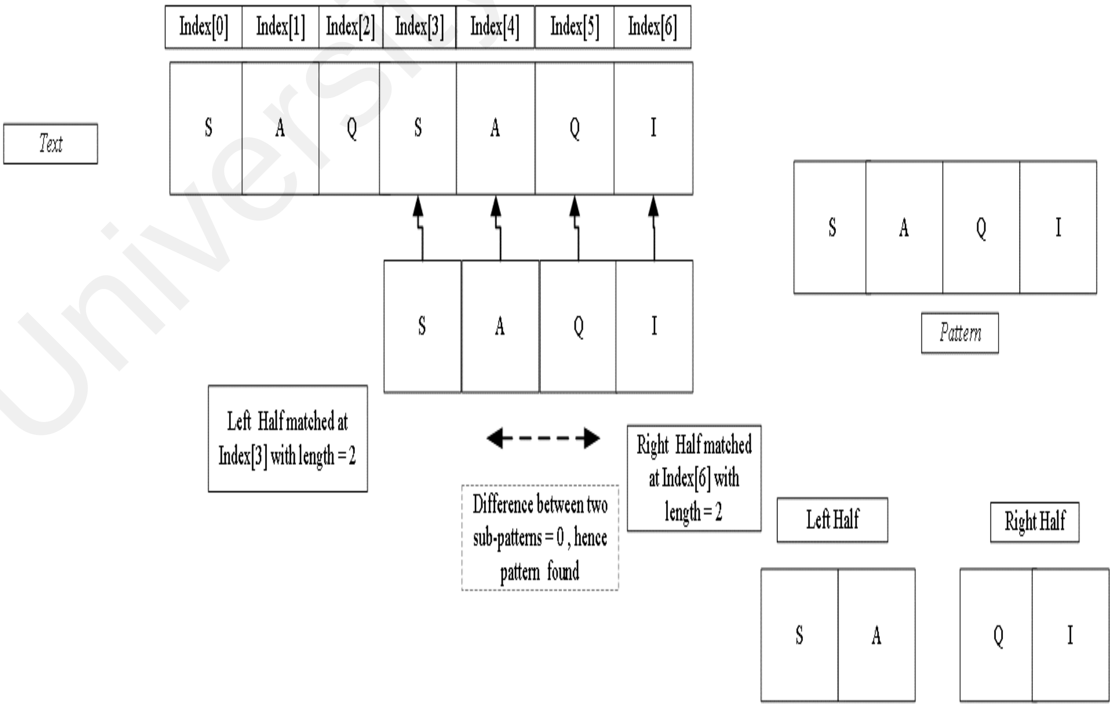

Index[0]
Index[1]
Index[2]
Index[3]
Index[4]
Index[5]
Index[6]
Text
S
A
Q
S
A
Q
I
S
A
Q
I
Left Half matched at Index[3] with length = 2
Difference between two sub-patterns = 0 , hence pattern found
Right Half matched at Index[6] with length = 2
S
A
Q
I
Pattern
Left Half
Right Half
S
A
Q
I

Figure 6.7: Example of a successful pattern search

The pseudo-code of the proposed algorithm is shown in Figure 6.8.

```
1.Input Pattern= " "
2.Given text = " "
3.Compute the length(L) of pattern
4.Compute the length(T) of text
5. Divide the pattern into two equal halves i.e. left and right with respective
length l and r
6.Compute l of left part and r of the right part
7.Search right part
        for ( i = 0; i <= T - r; i++)
            for (j = 0; j < r; j++) {
                if (text.charAt(i+j) != r.charAt(j))
                    break;
           if (j ==r)
Display (Right Pattern Found)
8. Search left part
        for ( i = 0; i <= T -l; i++)
            for (j = 0; j <l; j++) {
                if (text.charAt(i+j) != l.charAt(j))
                    break;
           if (j == l)
Display (Left Pattern Found)
9. Compute the differences between the locations of two parts i.e. the left and
right part.
If difference == 0 (pattern found)
```

Figure 6.8: Pseudo code of the partition-based algorithm

The logical steps of the proposed approach are shown in Figure 6.8, where from the lines 1-4, the length of the input pattern and the given text is determined. In lines 5-6, the input pattern is split into left and right halves respectively, and the length of each sub-part is calculated. The search phase of both halves is represented in lines 7 and 8, respectively. Finally, if there is a difference of zero between the right half and the left half, we obtain the expected result as represented in line 9. Otherwise, the whole process is repeated (lines 7-9).

6.5 Results

To evaluate the proposed approaches from sections 6.3 and 6.4, we consider a standard dataset of different natural texts, namely, English, Italian, Chinese, French and Arabic that Alsulami (Al-Ssulami, 2014) has taken for comparison purposes from the work of

Faro (Faro & Lecroq, 2013) and Tanzil.net (tanzil.net, 2016). It is noted that the Arabic and Chinese database use the UTF and GB18030 encoding scheme respectively to accommodate diacritics. Each Arabic character requires one-byte of information, while each Chinese character requires 2 bytes of information. The main reason to consider a dataset of different scripts is to show that the proposed approaches are script independent and occupy less amount of memory for all scripts. An experimental framework is now presented in Figure 6.9.

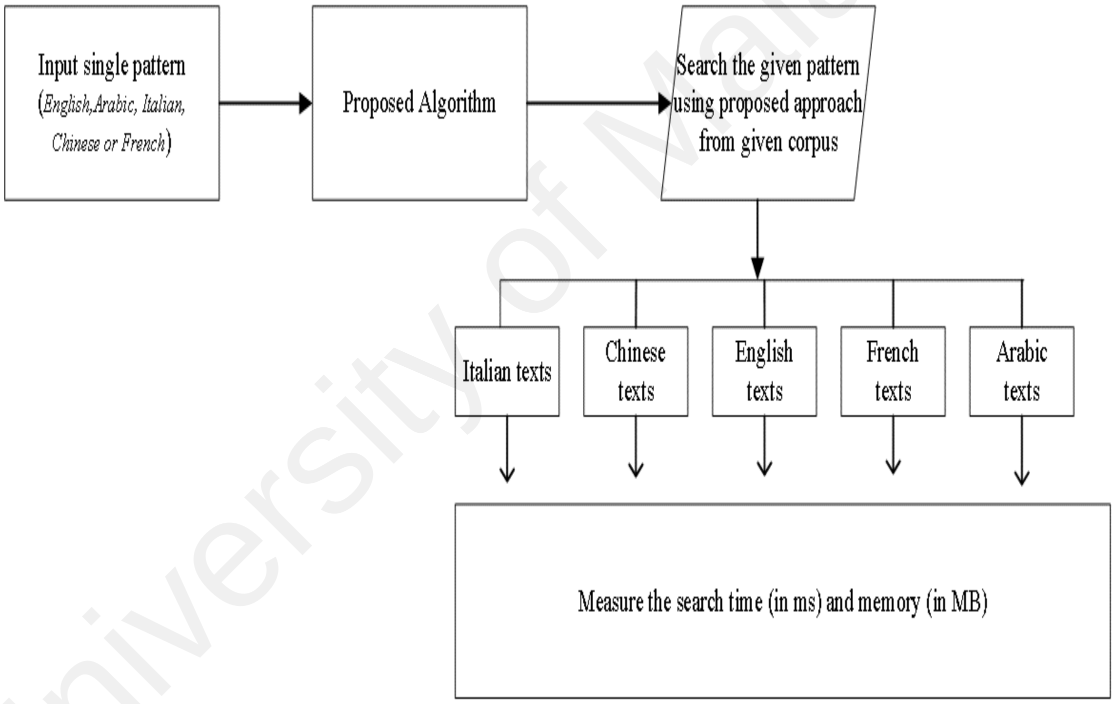

Input single pattern
(English,Arabic, Italian, Chinese or French)
Proposed Algorithm
Search the given pattern using proposed approach from given corpus
Italian texts
Chinese texts
English texts
French texts
Arabic texts
Measure the search time (in ms) and memory (in MB)

Figure 6.9: Experimental framework

Since our objective is to evaluate the proposed method in terms of time and space complexity, we use processing time in milliseconds and memory consumption in megabytes (MB) as our two performance measures. The same measures are used for all experimentations performed on the different script datasets.

To show the effectiveness and usefulness of the proposed approaches, we compare the results of the proposed algorithm with the results of well-known existing algorithms on different datasets. The existing algorithms are (1) Boyer-Moore (BM) algorithm, which considers the rightmost character of the pattern for searching and uses good-suffix shift and bad-character shift during matching, and (2), in which (Sunday, 1990) proposed an algorithm called BMT as an improved version of the BM algorithm. This algorithm combines the strengths of the BM and the KMP Knuth-Morris (KMP) algorithms. The basic idea behind the algorithm is that the text *t* is scanned from left to right, and when a mismatch occurs, the algorithm decides how much pattern *p* must be shifted to avoid redundant comparisons. Thus, it keeps track of information gained from previous comparisons. This algorithm skips characters based on the prefix and suffix rule (Knuth, 1977). Recently, a character based approach (SSM) proposed by (Al-Ssulami, 2014) compares the pivot character with the corresponding character and shifts the pattern either using Horspool shifts or hybrid shifts. Horspool proposed an algorithm known as the Horspool algorithm. This simplifies the Boyer-Moore's algorithm by dropping the good suffix rule. The shift is computed in such a way that rightmost character of the pattern becomes aligned with the rightmost occurrence in a given text (Horspool, 1980b). In the work of Faro (Faro & Lecroq, 2013), Hash 3 and SBNDM algorithms have shown better results among 85 algorithms for natural texts like Bible. For that reason, the SSM algorithm (Al-Ssulami, 2014) was compared with Hash 3 and SBNDM algorithms. SSM algorithm has demonstrated improved results compared to Hash 3 and SBNDM, indicating that the results are better than those existing 85 algorithms.

It is observed from the review of existing methods, that all four existing methods had used character components for matching and searching. Thus, it involves more computations, comparisons, and shifts, which results in more time processing and memory consumption. Furthermore, among the above existing methods, the traditional brute force criterion is common. Therefore, we also use the same criterion for our comparative study without additional features in this work.

For experimentation purposes, we consider a very short query pattern (1-3 character length) and medium query pattern (>=4 character length) to test the time efficiency of the proposed and existing algorithms using different datasets. As the main scope of this work is focused on improving the search process for Arabic texts, the reason for selecting a short pattern is due to the nature of the Arabic text, particularly Quranic verses. The Quranic verses are usually medium in length ranging from four characters per word (medium) to a minimum of two characters (short) along with diacritics as per our observations. Each algorithm is executed 10 times, and the execution times were calculated by taking the mean after executing the algorithm ten times.

The quantitative results of the proposed and existing algorithms for query pattern lengths on different datasets are reported in Tables 6.2 and 6.3, respectively. From Table 6.2, it is noticeable that the first proposed algorithm (A1) outperforms the existing algorithms for all the queries on Arabic and Chinese texts. Therefore, it can be argued that the proposed algorithm is effective for diacritical texts like Arabic in terms of time efficiency for very short patterns. Since the aim of the proposed algorithm is to achieve better time efficiency, it reports poor results for other datasets including Italian, English and French texts possibly due to the nature of those datasets with respect to the arrangement of selected patterns in case of very short patterns of size <4. However, in the case of medium patterns i.e. pattern length >4, the A1 approach performed better for Arabic and English texts only. For searching medium-sized Chinese texts, A1 showed poorer performance due to the brute force nature of the algorithm and complex nature of Chinese characters that take more than one byte per character.

As explained in section 6.3, to enhance the A1 approach, the A2 approach was proposed. Although the time complexity of the A2 is linear (i.e. *O (n)*), the proposed algorithm performed better as compared to the existing algorithms, due to the simplicity of the steps involved in pattern matching. In the existing pattern matching algorithms, the shifting process is the major cause of time consummation. This proposed approach does not include shifting phases, thus the T_L factor (from equation 1) is negligible and the search time considerably reduced. Besides, the use of modern appliances like i-5 or i-7 processors poses another advantage that processes bits faster, which further improves the searching process. However, the A2 approach showed poor performance in searching very short and medium-sized patterns for Chinese texts. This is again because each Chinese character takes 2 bytes per character. Hence, most of the very short and medium patterns consist of few characters only that result in poor searching.

Table 6.2: Processing time of the Proposed versus Existing Methods for Very Short Patterns in milli-seconds

Corpus	Text	Size (MB)	Sample patterns	BM	BMT	SSM	Brute Force	A1	A2
Arabic	Quran	0.7	بسم	1062	985	1235	2152	**937**	**855.3**
English	Bible	3.83	Good, the	5967	4200	5710	8527	4205	**2456**
Italian	Orlando	0.72	Dal, tal	5172	4058	4230	13430	4492	**3358.2**
Chinese	Journey	1.37	兒	3025	3969	2806	7547	**2500**	3031
French	L’homme	1.13	Dans, ils	3078	2284	2859	6995	2549	**2113.8**

Table 6.3: Processing time of the Proposed versus Existing Methods for Medium Patterns in milli-seconds

Corpus	Text	Size (MB)	Sample patterns	BM	BMT	SSM	Brute Force	A1	A2
Arabic	Quran	0.7	بسم لله رح	1094	1013	1078	2100	**844**	**793.8**
English	Bible	3.83	Continually, that Adam	2172	2253	2422	7939	**2092**	**2001**
Italian	Orlando	0.72	Trascorso, lungo tratto	4143	3937	3330	13648	4375	**3281**
Chinese	Journey	1.37	旗飛彩	**2083**	2093	2218	7655	2609	2156
French	L’hom me	1.13	Angleterre, imite le chinois	2869	2374	2719	7105	2384	**2129**

It must be noted that although exact matching algorithms search longer patterns quickly compared to smaller patterns, it is sometimes counter-intuitive. It depends on the location of the pattern within a given text. A short pattern occurring at the beginning of the given file will be searched quickly by the exact matching algorithm compared to long patterns that might be located at the end of the given file.

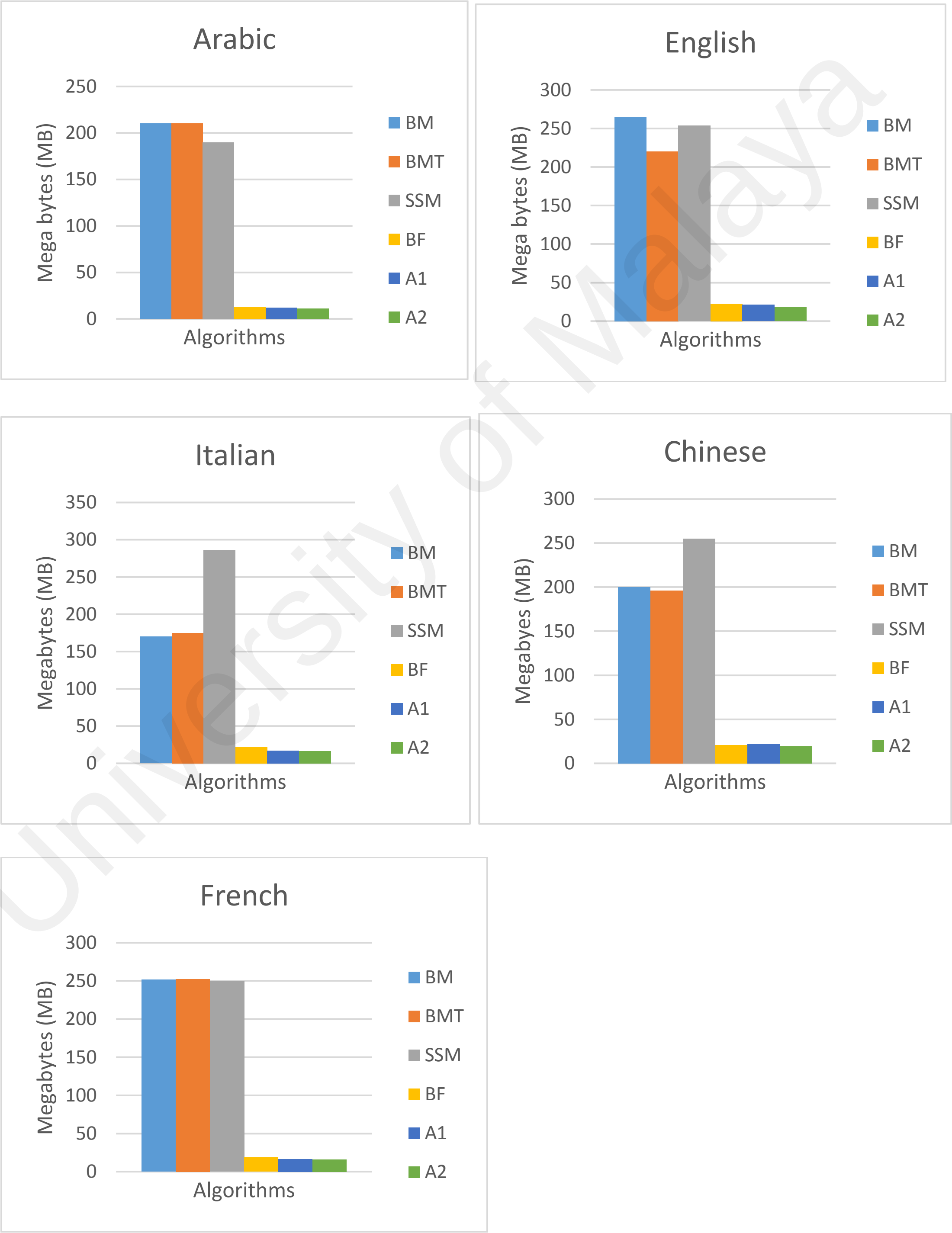
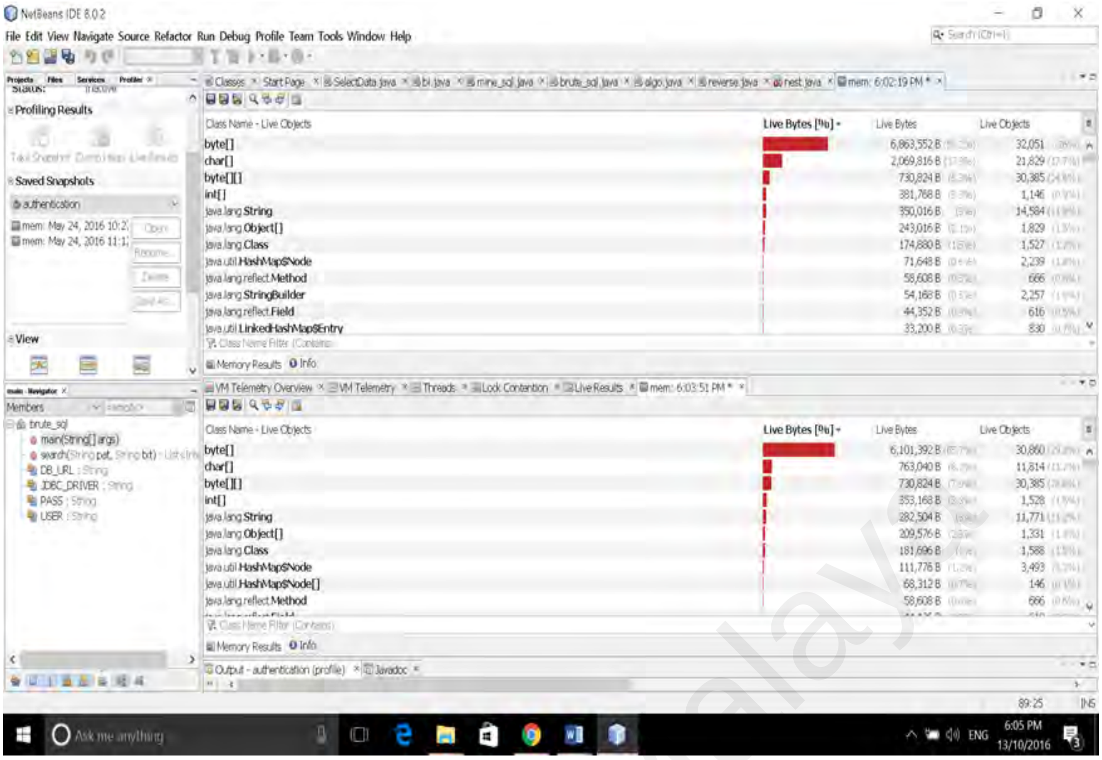


Figure 6.10: Memory analysis of string matching algorithms

Like those experiments that examine time efficiency for chosen datasets, we calculate the memory used for matching and searching of query words listed in Table 6.2-Table 6.3. The average memory consumption of different query words of the proposed and existing algorithms on different natural text datasets is shown in Figure 6.10. The figure shows that all the existing algorithms, except Brute force and the proposed approaches, consume 100 to 200 MB of heap memory during run-time. However, the brute force algorithm consumes less memory, i.e. less than 20 MB. Additionally, it requires more time for searching according to Table 6.2 and Table 6.3. Brute force algorithms require more operations for searching a string in the database. Since it requires more operations, usage of pointers, calling internal methods and variables also increase. Therefore, brute force algorithms consume more space than the proposed approaches. For a fair comparative study, we use the function in Java for estimating memory consumption for all the experiments shown in Figure 8. Memory requirements were analysed using a memory analysis tool available in java using Netbeans IDE (Oracle, 2018). From Figure 6.11, it can be seen the upper portion displays memory usage of the proposed approach and the lower part of the brute force algorithm. Although there is not much difference in terms of memory requirements (ranging from 15-25 MB) between the two. But, the slightest difference with respect to the memory requirements depends on the allocation of bytes and characters within the Java virtual machine (JVM) that occupy some of the memory. Therefore, we can confirm that the proposed approach achieves both time and space efficiency for different query pattern length on different natural text datasets.



Class Name - Live Objects	Live Bytes [%]	Live Bytes	Live Objects
byte[]		6,863,552 B	32,051
char[]		2,069,816 B	21,829
byte[][]		730,824 B	30,385
int[]		381,768 B	1,146
java.lang.String		350,016 B	14,584
java.lang.Object[]		243,016 B	1,829
java.lang.Class		174,880 B	1,527
java.util.HashMap$Node		71,648 B	2,239
java.lang.reflect.Method		58,608 B	666
java.lang.StringBuilder		54,168 B	2,257
java.lang.reflect.Field		44,352 B	616
java.util.LinkedHashMap$Entry		33,200 B	830

Class Name - Live Objects	Live Bytes [%]	Live Bytes	Live Objects
byte[]		6,101,392 B	30,860
char[]		763,040 B	11,814
byte[][]		730,824 B	30,385
int[]		353,168 B	1,528
java.lang.String		282,504 B	11,771
java.lang.Object[]		209,576 B	1,331
java.lang.Class		181,696 B	1,588
java.util.HashMap$Node		111,776 B	3,493
java.util.HashMap$Node[]		68,312 B	146
java.lang.reflect.Method		58,608 B	666

Figure 6.11: Brute Force Memory analysis

6.6 Summary

This chapter describes two proposed approaches for exact string matching to achieve both time and space efficiency for authentication purposes, regardless of query pattern lengths, dataset sizes and scripts. The proposed approaches split the given query pattern length into two halves. In the case of A1, only the right half is considered for searching in a text. Once the match is found for the right half, the proposed algorithm uses the left half extracted from the matched reference. This process helps reduce the number of computations (especially comparisons) whilst consuming less memory as no pre-processing is involved compared to existing exact matching algorithms. Following the A2 approach, both halves are searched simultaneously. Once both halves are located, the proposed approach computes the differences between them. A difference of zero between two halves indicates that the complete location of the pattern is being searched. In order to assess the usefulness of the proposed methods, experimental runs are conducted on various natural texts that include Arabic, English, Italian, French, and Chinese. The experimental results of the newly proposed and the already established algorithms on different script datasets, particularly Quranic texts, for different query pattern lengths show that the proposed methods outperform most of the existing algorithms in terms of time and space efficiency. Multiple verses consisting of short and medium patterns of DHQ are tested and evaluated using the proposed approaches and produce improved search time and memory consumption. Therefore, the proposed approaches are a script, query pattern length, and database size independent. As part of our future research, we plan to extend this approach for multiple patterns and implement the proposed approaches A1 and A2 using Graphical Processing Units (GPU).

CHAPTER 7: CONCLUSION

7.1 Introduction

This chapter discusses the overall work that was conducted to complete the proposed research. Firstly, the research objectives are revisited to explain the purpose and the outcome of this study. Secondly, the research contributions of the conducted research are highlighted. Finally, limitations of the conducted research along with future recommendations are given.

7.2 Research Objectives Revisited

This section revisits the research objectives of the conducted research.

7.2.1 Research Objective 1

The first objective reviews the state-of-the-art in authenticating sensitive diacritical text. Different approaches like pattern matching, watermarking, hashing and SQL based approaches are identified as the potential approaches that can be used for authenticating different sensitive diacritical content. The studies that explored steganography and cryptography for protecting the sensitive diacritical text, were also investigated. After conducting a prior investigation of all the relevant studies, the taxonomy based on preserving the content integrity of sensitive diacritical text is devised. The aim is to help upcoming researchers in the identification of proper authentication approaches.

7.2.2 Research Objective 2

The second objective is to authenticate two different writing styles using a single database. To achieve this objective, the logical operation based on XOR is carried out to find the difference between the two different text styles. The difference or residual helps substitute different characters using a simpler version of that particular character. The method is evaluated on DHQ using two popular writing styles of Uthmani and plain Quranic text. The detail discussion is presented in Chapter 4.

7.2.3 Research Objective 3

The third objective is to design a database in such a way that can solve the issue of authentication (search and retrieval) specifically of single texts. Upon the achievement of Objective 1 and 2, it is observed that most of the diacritical verses are not retrieved properly, and in some cases, fail to retrieve any result. Once again, the case study of DHQ is considered. Experiments performed on Quranic search engines reveal the same problem. Hence, the Quranic verses are arranged in a database based on their first characters. This additional measure improves the search and retrieval process considerably. The steps that are carried out to achieve this objective are presented in Chapter 5.

7.2.4 Research Objective 4

The final objective is to overcome the limitation of Objective 3 by authenticating more than one diacritical verse. For this purpose, exact matching (or pattern matching) approaches are explored. Finally, the split based approach is devised to solve the problem of authenticating multiple verses using two different algorithms. Different datasets including Arabic, English, French, Italian, and Chinese are taken for evaluation purposes. The details of this approach are provided in Chapter 6.

7.3 Research Contributions

The contribution of this research to the field of information security in terms of authentication (searching and retrieving) and text processing is rendered evident in the novel methods proposed for authenticating diacritical texts. Different issues related to content integrity authentication of sensitive content in Arabic, particularly diacritical content, is identified. Based on the issues related to content integrity authentication, different methods were proposed to search and authenticate complex diacritical texts to detect content-altering issues. The scope of studying diacritical texts is narrowed down to the case study of the Arabic Digital Quran given its complex grammatical nature and sensitivity to minute tampering. It thus provides a particularly suitable material for a case study.

A complete framework related to authentication and protection of diacritical text is proposed with the case study of DHQ. Since the scope of this study is limited to the authentication phase, four different objectives are identified as discussed above. The first objective conducts the comprehensive review related to the works done on sensitive diacritical texts. The focus of the second objective is to authenticate multiple-styled diacritical text by evaluating the method on DHQ, using Uthmani and plain script style. The scripts were authenticated by taking 1000 random verses using a single database with the accuracy of 87 %. To achieve this objective, an XOR-based approach is carried out to determine the differences between the two script styles and substitute the different letters with a common letter. After the completion of the second objective, it is observed that the presence of diacritics reduces the retrieval efficiency of diacritical texts. This inefficient retrieval slows down the process of authenticating any diacritical verse. To solve the issue of a poor retrieval, the third objective is formulated. It is also observed that the representation of sensitive diacritical text in the database constitutes one of the prime reasons for poor retrieval. This hypothesis is carried out using a factorial design technique to determine the influence of certain factors on the poor retrieval process. A new data representation method is proposed to achieve the third objective. Using the proposed data representation method, the verse is retrieved through the first verse character only, thus narrowing down the search process. The method is evaluated using DHQ. However, the solution proposed for objective 3 is not found suitable for retrieving more than one verse at a time. Hence, pattern matching algorithms are explored and split-based approaches are proposed to achieve the last objective.

These different methods can be used to develop a system that can authenticate diacritical texts based on the input, particularly digitalized versions of the sensitive diacritical text like Quranic text. The method proposed in Objective 3 can be used to authenticate a single verse. However, for authenticating the whole Quran or multiple verses in one search, the split-based approach proves to be the better choice.

All the findings from this research are published in academic journals and conference papers. Moreover, the novel method of authenticating the Digital Holy Quran has been patented with details given in the publications section.

7.4 Limitations and future work

The study area of sensitive diacritical texts and their authentication offers plenty of research opportunities. All the stages like the pre-processing and authentication phases possess certain limitations that can be studied, assessed and rectified. The limitations inherent in this study are highlighted below:

7.4.1 Limitation of Objective 2

Since the primary aim of the second objective is to authenticate two script styles of the diacritical text, the achieved accuracy falls short of 100 % in the case of DHQ. This is due to some complex words that cannot be changed due to their influence on the meaning of the verse. To improve the accuracy, new approaches like artificial intelligence can be explored. As our residual approach converts two most popular script styles of the Quranic text into a standard script style, other established script styles such as Kufi or Varsh can also be converted into a standard style for authentication.

7.4.2 Limitation of Objective 3

Similarly, the proposed data representation scheme explained above is limited to authenticating only one verse at a time. If there are two verses to be authenticated beginning with different letters, our approach cannot authenticate the second verse since it can only be located somewhere else in the database based on its first character. Hence, there is the potential to enhance the proposed representation that can authenticate multiple verses.

7.4.3 Limitation of Objective 4

Finally, the last two methods proposed based on exact matching algorithms demand more time when processing large databases due to the overhead associated in matching patterns from a given text character by character. However, it can be enhanced further using the concept of GPUs and parallel based approaches.

Future work in this area of research should include authenticating image-based sensitive diacritical versions, protecting sensitive diacritical content from tampering, authenticating Quranic recitation using the audio module and others. Other important future directions are listed in Chapter 2.

REFERENCES

Abdelali, A., Cowie, J., & Soliman, H. S. (2004, 19-22nd April, 2004). *Arabic information retrieval perspectives.* Paper presented at the Proceedings of the 11th Conference on Natural Language Processing (JEP-TALN).

Abuhaija, B., Shilbayeh, N., & Alwakeel, M. (2013, 22-24 June). *Security protocol architecture for website authentications and content integrity.* Paper presented at the World Congress on Computer and Information Technology (WCCIT).

Ahmad, M. K. (2014). *An Enhanced Boyer-Moore Algorithm (Doctoral dissertation).* Middle East University,Jordan,

Aho, A. V., & Corasick, M. J. (1975). Efficient string matching: an aid to bibliographic search. *Communications of the Acm, 18*(6), 333-340.

Al-Badarneh, A., Al-Shawakfa, E., Bani-Ismail, B., Al-Rababah, K., & Shatnawi, S. (2016). The impact of indexing approaches on Arabic text classification. *Journal of Information Science, 1*, 1-15.

Al-Dabbagh, S. S. M., Naser, M. A. S., & Barnouti, N. H. (2017). Fast Hybrid String Matching Algorithm based on the Quick-Skip and Tuned Boyer-Moore Algorithms. *International journal of advanced computer science and applications, 8*(6), 117-127.

Al-Sanabani, M., & Al-Hagree, S. (2016). Improved An Algorithm For Arabic Name Matching. *Open Transactions on Information Processing*, 1-17.

Al-Ssulami, A. M. (2014). Hybrid string matching algorithm with a pivot. *Journal of Information Science*, 82-88.

Al Gharaibeh, A., Al Taani, A., & Alsmadi, I. (2011). *The usage of formal methods in Quran search system.* Paper presented at the Proceedings of international conference on information and communication systems, Ibrid, Jordan.

AlAhmad, M. A., Alshaikhli, I., & Alduwaikh, A. E. (2013a). *A New Fragile Digital Watermarking Technique for a PDF Digital Holy Quran.* Paper presented at the International Conference on Advanced Computer Science Applications and Technologies (ACSAT).

AlAhmad, M. A., Alshaikhli, I., & Jumaah, B. (2013b). *Protection of the Digital Holy Quran Hash Digest by Using Cryptography Algorithms.* Paper presented at the International Conference on Advanced Computer Science Applications and Technologies (ACSAT).

Albujasim, Z. M. (2014). Search Queries in an Information Retrieval System for Arabic-Language Texts.

Alfaifi, A., & Atwell, E. (2016). Comparative evaluation of tools for Arabic corpora search and analysis. *International Journal of Speech Technology, 19*(2), 347-357.

Alfred, V. (2014). Algorithms for finding patterns in strings. *Algorithms and Complexity, 1*, 255-300.

Alginahi, Y. M., Tayan, O., & Kabir, M. N. (2013). Verification of Qur'anic Quotations Embedded in Online Arabic and Islamic Websites. *International Journal on Islamic Applications in Computer Science And Technology, 1*(2), 41-47.

Alhendawi, K. M., & Baharudin, A. S. (2013). String Matching Algorithms (SMAs): Survey & Empirical analysis. *Journal of Computer Sciences and Management*.

Almazrooie, M., Samsudin, A., Gutub, A. A.-A., Salleh, M. S., Omar, M. A., & Hassan, S. A. (2018). Integrity verification for digital Holy Quran verses using cryptographic hash function and compression. *Journal of King Saud University-Computer and Information Sciences*.

Alshareef, A., & Saddik, A. E. (2012). *A Quranic quote verification algorithm for verses authentication.* Paper presented at the International Conference on Innovations in Information Technology (IIT).

Alsmadi, I., & Zarour, M. (2015). Online integrity and authentication checking for Quran electronic versions. *Applied Computing and Informatics*, 1-16.

Arnold, M., Schmucker, M., & Wolthusen, S. D. (2002). *Techniques and applications of digital watermarking and content protection*: Artech House.

Arslan, A. (2015). DeASCIIfication approach to handle diacritics in Turkish information retrieval. *Information Processing & Management*, 326-339.

Atwan, J., Mohd, M., Rashaideh, H., & Kanaan, G. (2015). Semantically enhanced pseudo relevance feedback for Arabic information retrieval. *Journal of Information Science*, 1-15.

Bar-Ilan, J., & Gutman, T. (2005). How do search engines respond to some non-English queries? *Journal of Information Science, 31*(1), 13-28.

Bender, W., Gruhl, D., Morimoto, N., & Lu, A. (1996). Techniques for data hiding. *IBM systems journal, 35*(3.4), 313-336.

Bennett, K. (2004). *Linguistic steganography: Survey, analysis, and robustness concerns for hiding information in text*. Retrieved from Center for Education and Research in Information Assurance and Security,Purdue University,West Lafayette:

Berry, T., & Ravindran, S. (2001). *A Fast String Matching Algorithm and Experimental Results.* Paper presented at the Proceedings of the Prague Stringology Club Workshop'99.

Bobroff, N., Dawson, M. H., Fong, L. L., Iyengar, A. K., & Westerink, P. H. (2016). System and method for improving memory usage in virtual machines. In: US Patent 20,160,110,225.

Boyer, R. S., & Moore, J. S. (1977). A fast string searching algorithm. *Communications of the Acm, 20*(10), 762-772.

Brassil, J. T., Low, S., & Maxemchuk, N. F. (1999). *Copyright protection for the electronic distribution of text documents.* Paper presented at the Proceedings of the IEEE.

Chang, A. X., & Manning, C. D. (2014). TokensRegex: Defining cascaded regular expressions over tokens. In *Technical Report CSTR 2014-02*: Department of Computer Science, Stanford University.

Charras, C., & Lecroq, T. (2004). *Handbook of exact string matching algorithms*: H. King's College Publications.

Cole, E. (2003). Hiding in plain sight. *Steganography and the Art of Covert Communication, Wiley*.

Coron, J.-S. (2006). What is cryptography? *IEEE security & privacy, 4*(1), 70-73.

Crochemore, M., Czumaj, A., Gasieniec, L., Jarominek, S., Lecroq, T., Plandowski, W., & Rytter, W. (1994). Speeding up two string-matching algorithms. *Algorithmica*, 247-267.

Daraee, F., & Mozaffari, S. (2014). Watermarking in binary document images using fractal codes. *Pattern Recognition Letters, 35*, 120-129.

Darwish, K., & Magdy, W. (2014). *Arabic information retrieval*: Now Publishers.

Date, C. J., & Darwen, H. (1997). *A Guide to the SQL Standard: A User's Guide to the Standard Database Language SQL* (4 ed.). Boston, MA, USA: Addison-Wesley.

Delfs, H., & Knebl, H. (2015). Symmetric-Key Cryptography. In *Introduction to Cryptography* (pp. 11-48): Springer.

El-Defrawy, M., El-Sonbaty, Y., & Belal, N. A. (2016). A Rule-Based Subject-Correlated Arabic Stemmer. *Arabian Journal for Science and Engineering, 41*(8), 2883-2891.

Elayeb, B., & Bounhas, I. (2016). Arabic Cross-Language Information Retrieval: A Review. *Acm Transactions on Asian and Low-Resource Language Information Processing, 15*(3), 18.

Fadi, S. (2017). www.arabion.net. Retrieved from http://www.arabion.net/lesson4.html

Farghaly, A., & Shaalan, K. (2009). Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP), 8*(4), 14.

Faro, S., & Külekci, M. O. (2013). *Fast packed string matching for short patterns.* Paper presented at the Proceedings of the Meeting on Algorithm Engineering & Expermiments.

Faro, S., & Lecroq, T. (2013). The exact online string matching problem. *Acm Computing Surveys, 45*(2), 1-42.

Feller, W. (1968). *An introduction to probability theory and its applications: volume I* (Vol. 3): John Wiley & Sons London-New York-Sydney-Toronto.

Frakes, W. B., & Baeza-Yates, R. (1992). *Information retrieval: data structures and algorithms* (Vol. 331): Englewood Cliffs, NJ: prentice Hall.

Greenspan, J., & Bulger, B. (2001). *MySQL/PHP database applications*: John Wiley & Sons, Inc.

gs.statcounter. (2018). Desktop vs Mobile vs Tablet Market Share Worldwide. Retrieved from http://gs.statcounter.com/platform-market-share/desktop-mobile-tablet.

Gutub, A. A.-A., Al-Alwani, W., & Mahfoodh, A. B. (2010). Improved method of Arabic text steganography using the extension 'Kashida'character. *Bahria University Journal of Information & Communication Technology, 3*(1), 68-72.

Hakak, S., Kamsin, A., Palaiahnakote, S., Tayan, O., Idris, M. Y. I., & Abukhir, K. Z. (2018a). Residual-based approach for authenticating pattern of multi-style diacritical Arabic texts. *PloS one, 13*(6).

Hakak, S., Kamsin, A., Tayan, O., Idris, M. Y. I., Gani, A., & Zerdoumi, S. (2017a). Preserving content integrity of digital holy Quran: Survey and open challenges. *Ieee Access, 5*, 7305-7325.

Hakak, S., Kamsin, A., Tayan, O., Idris, M. Y. I., & Gilkar, G. A. (2017b). Approaches for preserving content integrity of sensitive online Arabic content: A survey and research challenges. *Information Processing & Management*.

Hakak, S., Kamsin, A., Veri, J., Ritonga, R., & Herawan, T. (2018b). A Framework for Authentication of Digital Quran. In *Information Systems Design and Intelligent Applications* (pp. 752-764): Springer.

Hakak, S., Latif, S. A., Anwar, F., Alam, M. K., & Gilkar, G. (2014). Effect of 3 Key Factors on Average End to End Delay and Jitter in MANET. *Journal of ICT Research and Applications*, 113-125.

Hakak, S. I., Kamsin, A., Idris, M. Y. I., Gani, A., Amin, G., & Zerdoumi, S. (2017c). Diacritical Digital Quran Authentication Model. *Science and Technology, 25*, 133-142.

Hamariweb. (2018). www.hamariweb.com. In. Retrieved from http://hamariweb.com/poetries/default.aspx.

Hammo, B., Sleit, A., & El-Haj, M. (2007). *Effectiveness of query expansion in searching the Holy Quran.* Paper presented at the The Second International Conference on Arabic Language Processing, Morocco.

Hammo, B. H. (2008). Towards enhancing retrieval effectiveness of search engines for diacritisized Arabic documents. *Information retrieval, 12*(3), 300-323.

Haouzia, A., & Noumeir, R. (2008). Methods for image authentication: a survey. *Multimedia Tools and Applications, 39*(1), 1-46.

Hassan, T., Wassim, A.-F., & Bassem, M. (2007). *Analysis and Implementation of an Automated Delimiter of" Quranic" Verses in Audio Files using Speech Recognition Techniques*: INTECH Open Access Publisher.

Hennessy, J. L., Patterson, D. A., & Larus, J. R. (1999). *Computer organization and design: the hardware/software interface (3rd print. ed.). San Francisco: Kaufmann*. ISBN 155860-428-6.

Hlayel, A. A., & Hnaif, A. (2014). An algorithm to improve the performance of string matching. *Journal of Information Science, 40*(3), 357-362.

Horspool, R. N. (1980a). Practical fast searching in strings. *Software: Practice and Experience, 10*(6), 501-506.

Horspool, R. N., 10(6), 501-506. (1980b). Practical fast searching in strings. *Software: Practice and Experience*, 501-506.

Hudaib, A., Al-khalid., D,Suleiman., Itriq,M., & Al-anani,Al. (2008). A fast pattern matching algorithm with two sliding windows (TSW). *Journal of Computer Science, 4*(5), 393-401.

Hume, A., & Sunday, D. (1991). Fast string searching. *Software: Practice and Experience, 21*(11), 1221-1248.

Ibrahim, N. J. (2010). *Automated TAJWEED checking rules engine for Quranic verse recitation.* (Doctoral dissertation), University of Malaya,

Ilie, L. (2008). Regular Expression Matching. In *Encyclopedia of Algorithms* (pp. 1-99): Springer.

Internet World Stats. (2018). www.internetworldstats.com. Retrieved from http://www.internetworldstats.com/stats.htm

Ismail, A., Idris, M. Y. I., Noor, N. M., Razak, Z., & Yusoff, Z. (2014). Mfcc-Vq Approachfor Qalqalah Tajweed Rule Checking. *Malaysian Journal of Computer Science, 27*(4), 275-293.

Jain, R. (1991). *The Art of Computer Systems Performance.*: Wiley.

Jaiswal, M. (2014). Accelerating Enhanced Boyer-Moore String Matching Algorithm on Multicore GPU for Network Security. *International Journal of Computer Applications, 97*(1).

Jensen, M., Schwenk, J., Gruschka, N., & Iacono, L. L. (2009). *On Technical Security Issues in Cloud Computing*. Paper presented at the 2009 IEEE International Conference on Cloud Computing.

Kamsin, A., Gani, A., Suliaman, I., Jaafar, S., Mahmud, R., Sabri, M., . . . Ismail, M. A. (2014). *Developing the novel Quran and Hadith authentication system.* Paper presented at the The 5th International Conference on Information and Communication Technology for The Muslim World (ICT4M).

Kanan, T., & Fox, E. A. (2016). Automated arabic text classification with P-Stemmer, machine learning, and a tailored news article taxonomy. *Journal of the Association for Information Science and Technology*.

Karp, R. M., & Rabin, M. O. (1987). Efficient randomized pattern-matching algorithms. *IBM Journal of Research and Development, 31*(2), 249-260.

kathir, D. b. (2017). *Holy Quran - Uthmani-Kaloon* (Vol. 1427). Damascus.

Katzenbeisser, S., & Petitcolas, F. (2000). *Information hiding techniques for steganography and digital watermarking*: Artech house.

Khalaf, E., F, D. K., & Morfeq, A. (2014). Arabic Vowels Recognition by Modular Arithmetic and Wavelets using Neural Network. *Life Science Journal, 11*(3), 33-41.

Khalil, M. S., Kurniawan, F., Khan, M. K., & Alginahi, Y. M. (2014). Two-layer fragile watermarking method secured with chaotic map for authentication of digital Holy Quran. *ScientificWorldJournal, 2014*, 803983.

Kim, J.-S., & Hsu, Y. (2000). *Memory system behavior of Java programs: methodology and analysis.* Paper presented at the ACM SIGMETRICS Performance Evaluation Review.

Kirchhoff, K., & Vergyri, D. (2005). Cross-dialectal data sharing for acoustic modeling in Arabic speech recognition. *Speech Communication, 46*(1), 37-51.

Knuth, D. E., Morris, Jr, J. H., & Pratt, V. R. (1977). Fast pattern matching in strings. *SIAM journal on computing*, 323-350.

Kurniawan, F., Khalil, M. S., Khan, M. K., & Alginahi, Y. M. (2013a). *Authentication and Tamper Detection of Digital Holy Quran Images.* Paper presented at the International Symposium on Biometrics and Security Technologies (ISBAST), 2013.

Kurniawan, F., Khalil, M. S., Khan, M. K., & Alginahi, Y. M. (2013b). Exploiting Digital Watermarking to Preserve Integrity of The Digital Holy Quran Images. *2013 Taibah University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences*, 30-36.

Kurniawan, F., Khalil, M. S., Khan, M. K., & Alginahi, Y. M. (2014). *DWT+ LSB-based fragile watermarking method for digital Quran images.* Paper presented at the International Symposium on Biometrics and Security Technologies (ISBAST).

Laouamer, L., & Tayan, O. (2013). An enhanced SVD technique for authentication and protection of text-images using a case study on digital Quran content with sensitivity constraints. *Life Science Journal, 10*(2), 2591-2597.

Lecroq, T. (2007). Fast exact string matching algorithms. *Information Processing Letters, 102*(6), 229-235.

Lin, C.-H., Liu, C.-H., Chien, L.-S., & Chang, S.-C. (2013). Accelerating pattern matching using a novel parallel algorithm on GPUs. *IEEE Transactions on Computers, 62*(10), 1906-1916.

Lin, J., Adjeroh, D., & Jiang, Y. (2014). A Faster Quick Search Algorithm. *Algorithms, 7*(2), 253-275.

Makbol, N. M., Khoo, B. E., & Rassem, T. H. (2016). Block-based discrete wavelet transform-singular value decomposition image watermarking scheme using human visual system characteristics. *IET Image Processing, 10*(1), 34-52.

Mano, M. M. (2017). *Digital logic and computer design*: Pearson Education India.

McEnery, A., & Xiao, R. (2005). *Character encoding in corpus construction*. Retrieved from http://ota.ox.ac.uk/documents/creating/dlc/chapter1.htm

McEnery, T., Xiao, R., & Tono, Y. (2000). Corpus-based language studies: An advanced resource book. In: Taylor & Francis.

Metwally, A. S., Rashwan, M. A., & Atiya, A. F. (2016). *A multi-layered approach for Arabic text diacritization.* Paper presented at the IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA).

Michailidis, P. D., & Margaritis, K. G. (2002). On-line approximate string searching algorithms: Survey and experimental results. *International Journal of Computer Mathematics, 79*(8), 867-888.

Mitali, V. K., & Sharma, A. (2014). A Survey on Various Cryptography Techniques. *International Journal of Emerging Trends & Technology in Computer Science, 3*(4), 307-312.

Mohammed, A., Sunar, M. S., & Salam, M. S. H. (2015). Quranic Verses Verification using Speech Recognition Techniques. *Jurnal Teknologi, 73*(2), 99-106.

Moukdad, H. (2013). *Lost in Cyberspace: How do search engines handle Arabic queries?* Paper presented at the Proceedings of the Annual Conference of CAIS/Actes du congrès annuel de l'ACSI.

Muslim-web. (2018). www.quran.muslim-web.com. Retrieved from http://quran.muslim-web.com.

Navarro, G. (2001). A guided tour to approximate string matching. *Acm Computing Surveys, 33*(1), 31-88.

Nin, J., & Ricciardi, S. (2013). *Digital watermarking techniques and security issues in the information and communication society.* Paper presented at the International Conference on Advanced Information Networking and Applications Workshops (WAINA).

Nisha, S., Ali, N., & Shawkat Ali, A. (2014). *Searching quranic verses: A keyword based query solution using. net platform.* Paper presented at the The 5th International

Conference on Information and Communication Technology for The Muslim World (ICT4M).

Nsira, N. B., Lecroq, T., & Elloumi, M. (2015). A fast Boyer-Moore type pattern matching algorithm for highly similar sequences. *International journal of data mining and bioinformatics, 13*(3), 266-288.

Nuaymi, L. (2007). *WiMAX: technology for broadband wireless access*: John Wiley & Sons.

Oracle. (2018). Netbeans. Retrieved from https://profiler.netbeans.org/docs/help/5.5/results_objliveness.html.

Pan, J.-S., Huang, H.-C., & Jain, L. C. (2004). *Intelligent watermarking techniques* (Vol. 7): World scientific.

Pinkerton, B. (2000). *Webcrawler: Finding what people want*: Citeseer.

Rafe, V., & Nozari, M. (2014). An Efficient Indexing Approach to Find Quranic Symbols in Large Texts. *Indian Journal of Science and Technology, 7*(10), 1643-1649.

Rafiq, A. N. M. E., El-Kharashi, M. W., & Gebali, F. (2004). A fast string search algorithm for deep packet classification. *Computer Communications, 27*(15), 1524-1538.

Rahim, R., Ahmar, A. S., Ardyanti, A. P., & Nofriansyah, D. (2017). *Visual Approach of Searching Process using Boyer-Moore Algorithm.* Paper presented at the Journal of Physics: Conference Series.

Rahman, T. F. A., Buja, A. G., Abd, K., & Ali, F. M. (2017). SQL Injection Attack Scanner Using Boyer-Moore String Matching Algorithm. *JCP, 12*(2), 183-189.

Raita, T. (1992). Tuning the boyer-moore-horspool string searching algorithm. *Software: Practice and Experience*, 879-884.

Religion of peace. (2018). What makes Islam so different,. Retrieved from https://www.thereligionofpeace.com/pages/quran/violence.aspx

Ryding, K. C. (2005). *A reference grammar of modern standard Arabic*: Cambridge university press.

Sabbah, T., & Selamat, A. (2013). *A framework for Quranic verses authenticity detection in online forum.* Paper presented at the Taibah University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences.

Sabbah, T., & Selamat, A. (2014). *Support vector machine based approach for quranic words detection in online textual content.* Paper presented at the 8th Malaysian Software Engineering Conference (MySEC).

Sabbah, T., & Selamat, A. (2015). A Novel Dataset for Quranic Words Identification and Authentication. *Jurnal Teknologi, 75*(2), 125-131.

Saleh, A. Z. M., Rozali, N. A., Buja, A. G., Jalil, K. A., Ali, F. H. M., & Rahman, T. F. A. (2015). A method for web application vulnerabilities detection by using boyer-moore string matching algorithm. *Procedia Computer Science, 72*, 112-121.

Schellenkens, M. H. M. (2004). *Electronic Signatures Authentication Technology from a Legal Perspective*: TMC Asser Press.

Searchtruth.com. (2018). http://www.searchtruth.com/. Retrieved from http://www.searchtruth.com/.

Shaham, R., Kolodner, E. K., & Sagiv, M. (2001). *Heap profiling for space-efficient Java.* Paper presented at the ACM SIGPLAN Notices.

Singh, P., & Chadha, R. (2013). A survey of digital watermarking techniques, applications and attacks. *International Journal of Engineering and Innovative Technology (IJEIT), 2*(9), 165-175.

Sri, M. B., Bhavsar, R., & Narooka, P. (2018). String Matching Algorithms. *International Journal Of Engineering And Computer Science, 7*(03), 23769-23772.

Strötgen, J., Armiti, A., Van Canh, T., Zell, J., & Gertz, M. (2014). Time for more languages: Temporal tagging of Arabic, Italian, Spanish, and Vietnamese. *ACM Transactions on Asian Language Information Processing (TALIP), 13*(1), 1.

Sumathi, C., Santanam, T., & Umamaheswari, G. (2014). A Study of Various Steganographic Techniques Used for Information Hiding. *arXiv preprint arXiv:1401.5561*.

Sunday, D. M. (1990). A very fast substring search algorithm. *Communications of the ACM, 33*(8), 132-142.

Syifak Izhar, H., Jasni, M. Z., Afifah Nailah, M., & Gran, B. (2013). Localization Watermarking for Authentication of Text Images in Quran.

tanzil.net. (2016). www.tanzil.net. Retrieved from http://tanzil.net/#2:1.

Tao, H., Chongmin, L., Zain, J. M., & Abdalla, A. N. (2014). Robust image watermarking theories and techniques: A review. *Journal of applied research and technology, 12*(1), 122-138.

Tayan, O., Alginahi, Y. M., & Kabir, M. N. (2013). *An Adaptive Zero-Watermarking Approach for Text Documents Protection*. Paper presented at the International Conference on Advances in Computer and Information Technology.

Tayan, O., Kabir, M. N., & Alginahi, Y. M. (2014). A hybrid digital-signature and zero-watermarking approach for authentication and protection of sensitive electronic documents. *ScientificWorldJournal, 2014*, 514652.

Techtarget. (2017). Hashing. Retrieved from http://searchsqlserver.techtarget.com/definition/hashing.

Welling, L., & Thomson, L. (2003). *PHP and MySQL Web development*: Sams Publishing.

Wong, K. K.-H. (2016). *Tweaking generic OTR to avoid forgery attacks.* Paper presented at the Applications and Techniques in Information Security: 6th International Conference Proceedings, ATIS 2016, Cairns, QLD, Australia, October 26-28.

WoS. (2018). web of science. Retrieved from https://clarivate.com/products/web-of-science/.

Xuehua, J. (2010). *Digital watermarking and its application in image copyright protection.* Paper presented at the International Conference on Intelligent Computation Technology and Automation (ICICTA).

Yang, T., Hertz, M., Berger, E. D., Kaplan, S. F., & Moss, J. E. B. (2004). *Automatic heap sizing: Taking real memory into account.* Paper presented at the Proceedings of the 4th international symposium on Memory management.

Yuan, J., Zheng, J., & Ding, S. (2010). *An improved pattern matching algorithm.* Paper presented at the Third International Symposium on Intelligent Information Technology and Security Informatics (IITSI).

Zerdoumi, S., Sabri, A. Q. M., Kamsin, A., Hashem, I. A. T., Gani, A., Hakak, S., . . . Chang, V. (2017). Image pattern recognition in big data: taxonomy and open challenges: survey. *Multimedia Tools and Applications*, 1-31.

LIST OF PUBLICATIONS AND PAPERS PRESENTED

Patents Filed:

- System and Method for Authenticating Religious Content Using a Removable Data Transfer Device (P.I - 2016002218).
- System and Method for Authenticating Quranic content and Hadith (P.I-2016001134).

ISI-Indexed papers:

- **Saqib Hakak,** Amirrudin Kamsin, Omar Tayan, Mohd Yamani Idna Idris, Gulshan Amin Gilkar, *Approaches for preserving content integrity of sensitive online Arabic content: A survey and research challenges*, **Information Processing & Management**, Available online 23 **August 2017**, ISSN 0306-4573, https://doi.org/10.1016/j.ipm.2017.08.004. (ISI-Indexed, Q1) - published.
- **Hakak, Saqib**, Amirrudin Kamsin, Omar Tayan, Mohd Yamani Idna Idris, Abdullah Gani, and Saber Zerdoumi. "*Preserving Content Integrity of Digital Holy Quran: Survey and Open Challenges.*" **IEEE Access (2017)**. (ISI-Indexed, Q1) – published.
- **Saqib Hakak**, Amirrudin.K, Yamani.M, Abdullah Gani "*Partition Based Pattern Matching Approach for Efficient Retrieval of Arabic Text* "**Malaysian Journal of Computer Science (2018)**. (ISI-Indexed, Q4) -(Accepted).
- **Saqib Hakak**, Amirrudin Kamsin, Mohd Yamani Idna Idris, Tayan.O "*An Efficient Data Structure Representation for Text Extraction from Arabic/Quran/Farsi Documents* "**Procedia Computer Science**". ISI-Indexed, (under review).

- **Saqib Hakak,** Amirrudin.K, Yamani.M, Tayan.O .“Residual based Approach for Authenticating Pattern of Multi-Style Vowelized Arabic Texts “**PLOS ONE (2018)”** published.
- **Saqib Hakak**, Amirrudin.K, Yamani.M, Abdullah Gani "*A new split-based exact Matching Approach for searching texts* “**PLOS ONE (2018).** (ISI-Indexed, Q1) (Accepted).
- Zerdoumi Saber, Aznul Qalid Md Sabri, Amirrudin Kamsin, **Saqib Hakak**, Abdullah Gani (2017). *Image Pattern Recognition in Big Data: Taxonomy and Open Challenges: Survey*. **Multimedia Tools and Applications**. (ISI-Indexed, Q2) - published.

Scopus/Conference papers:

- **Saqib Hakak**, Amirrudin.K, Yamani.M, Gulshan Amin Gilkar, Saber Zerdoumi and Abdullah Gani “Diacritical Quran Authentication model” *Pertanika Journal of Science and Technology* **(2018)** – (Accepted).
- **Saqib Hakak**, Amirrudin.K, Yamani.M, Saber Zerdoumi “A Framework for Authentication of Digital Quran” *Advances in Intelligent Systems and Computing AISC Series* **(Springer) (2018)** – (Accepted).
- **Saqib Hakak,** Amirrudin.K, Yamani.M, Abdullah Gani “*Identification of Techniques Suitable for preserving Content Integrity of Digital Quran*” Proceedings of National Symposium on Al-Quran and Hadith Validation System (SAHIH-2016). 15-16 March 2016, Kuala Lumpur, Malaysia. (Accepted).

Awards:

- **Saqib Hakak**, Amirrudin Kamsin, Mohd. Yamani idna Idris, Saber Zerdoumi and Abdullah Gani "*System and method for authenticating Quran*" **PECIPTA 2017**"- 7th-9th October 2017. (**Winner of SPECIAL AWARD & GOLD)**
- **Saqib Hakak**, Amirrudin Kamsin, Mohd. Yamani idna Idris, Saber Zerdoumi and Abdullah Gani "*System and method for authenticating Quran*" **IIDEX 2017**"- 26th-29th September 2017. (**Winner of GOLD**)
- **Saqib Hakak**, Amirrudin Kamsin, Mohd. Yamani idna Idris, and Abdullah Gani "*Digital Quran authentication system*" **ITEX 2018**"- 10th-29th May 2018. (**Winner of GOLD**).