IMPROVED PREPROCESSING AND CLASSIFICATION
ALGORITHMS FOR ONLINE SIGNATURE VERIFICATION

ELŐFELDOLGOZÁSI ÉS OSZTÁLYOZÁSI ALGORITMUSOK
TELJESÍTMÉNYÉNEK JAVÍTÁSA AZ ONLINE
ALÁÍRÁSHITELESÍTÉSBEN

Ph.D. Thesis

Mohammad Majed Ahmad Saleem

Advisor: Bence Kovari, Associate Professor

2021
Preface

Declaration of own work and references

I, undersigned Mohammad Saleem hereby declare that this Ph.D. dissertation was made by myself, and I only used the sources given at the end. Every part that was quoted word-for-word, or was taken over with the same content, I noted explicitly by giving the reference to the source.

Budapest, 2021.

(Mohammad Saleem)
Abstract

Handwritten signature is one of the most popular and used biometrics for authentication and verification. As technology improved, the use of digital signatures rapidly increased in the society such as in bank checks. Over the past few decades, researchers adopted various approaches by incorporating the latest image and signal processing technologies into their verification systems to achieve the maximum possible accuracy, taking into account both errors types of false accepted and false rejected signatures.

The main steps of any verification system are data acquisition, preprocessing, feature extraction, and verification. Although most of the system investigates the results of the system accuracy, there is a lack of research’s that studies the particular algorithms used in each step or compare them.

Therefore, the first part of this thesis addresses this problem by conducting a systematic evaluation of the most common pre-processing techniques used in conjunction with dynamic time warping-based classification. In addition, the results of this study are used to build a competitive online signature verification system.

As next, a new approach of using signature down-sampling and signer-dependent sampling frequency approach is proposed. This approach shows promising results in the field comparing to the literature.

After that, optimized and combined verification algorithms are used to propose real-life scenario online signature verification system. Some of the proposed methods took part in a well-reputed competition in the field.

In order to demonstrate the practical applications of the results, a complete signature verification benchmark has been developed, which incorporates the applied algorithms beside dozens of existing algorithms in the field and made public to help the researchers in the field.
Acknowledgments

Throughout the time this dissertation was in the making, and the years of research preceding it, I had the good fortune to receive support from my family and a great number of colleagues and friends.

First and foremost, I would like to say heartfelt thanks to my family for the long years of continuous support. I dedicate this dissertation to my lovely and amazing parents, who were always the main reason I kept going on and trying to be my best. Thank you for all your support and love. And to my great brothers for their amazing support and encouragement. And to the lovely new members of the family, Alma, Malak, and Asir.

Some special words of gratitude go to my friends who have always been a major source of support, especially to the ones who shared this long journey with me.

I’d like to express my gratitude towards my advisor for his guidance, encouragement, and patience throughout my research. I also want to thank the Stipendium Hungarian for funding my research.
# Table of Contents

Acknowledgments iv  
List of Figures ix  
List of Tables xi  

## Chapter 1  
Introduction 1  
1.1 Thesis contributions 3  
1.2 Thesis structure 4  

## Chapter 2  
Background and related work 6  
2.1 Terms and concepts 6  
2.2 Performance metrics 7  
2.3 Background 8  
2.4 Verification process and related work 9  
2.4.1 Data acquisition 10  
2.4.2 Databases 11  
2.4.2.1 SVC2004 11  
2.4.2.2 MCYT 12  
2.4.2.3 SigComp’11 12  
2.4.2.4 SigWiComp2015 German 12  
2.4.2.5 SUSIG 12  
2.4.2.6 BioSecure 13  
2.4.2.7 SigWiComp2013 13  
2.4.2.8 SIGMA 13
2.4.2.9 DeepSignDB ........................................ 14
2.4.2.10 SVC2021_EvalDB ................................. 14
2.4.2.11 ATVs ............................................ 15
2.4.2.12 Own databases .................................... 15
2.4.3 Preprocessing ....................................... 15
2.4.4 Feature extraction ................................... 19
2.4.5 Similarity Measurements and Classification ........... 19
2.5 State of the art results comparison .................. 23

Chapter 3
Systematic evaluation of preprocessing approaches in online signature verification 25
3.1 Methodology ........................................... 26
3.1.1 Preprocessing .................................... 26
3.1.2 Dynamic Time Warping (DTW) ....................... 26
3.1.3 Translation ...................................... 27
3.1.4 Scaling ........................................... 28
3.1.5 Rotation normalization .............................. 30
3.1.6 Pen-up strokes .................................. 30
3.1.7 Resampling ...................................... 31
3.2 Experimental setup ................................... 32
3.2.1 Datasets .......................................... 33
3.2.2 Feature selection ................................ 33
3.2.3 Test set selection (split) ........................... 34
3.2.4 Classification ................................... 34
3.3 Experimental results .................................. 35
3.3.1 Effects of training set selection ..................... 35
3.3.2 Effects of resampling ............................... 36
3.3.3 Handling of pen-up strokes ......................... 37
3.3.4 Distance function within DTW ...................... 38
3.3.5 Effects of translation and scaling ................... 39
3.3.6 The effects of rotation normalization ................ 40
3.3.7 Feature selection ................................ 42
3.3.8 Proposed system(s): ............................... 43
3.4 Chapter summary .................................... 44

Chapter 4
Sampling frequency based online signature verification system 47
4.1 The effect of sampling rate on the verification accuracy . . . . . . . 47
  4.1.1 Related work . . . . . . . . . . . . . . . . . . . . . . . . . . 48
  4.1.2 Proposed verification system . . . . . . . . . . . . . . . . . 50
  4.1.3 Signature sampling . . . . . . . . . . . . . . . . . . . . . . 52
  4.1.4 Experimental protocol . . . . . . . . . . . . . . . . . . . . . 52
  4.1.5 Experimental results . . . . . . . . . . . . . . . . . . . . . . 53
    4.1.5.1 EER expected vs. actual behavior . . . . . . . . . 53
    4.1.5.2 The best frequency ranges . . . . . . . . . . . . . . 56
    4.1.5.3 The best sample count ranges . . . . . . . . . . . . 56
    4.1.5.4 Sampling restrictions . . . . . . . . . . . . . . . . 57
    4.1.5.5 Comparison . . . . . . . . . . . . . . . . . . . . . 58
4.2 Signer-dependent sampling frequency approach for signature verifi-
    cation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
  4.2.1 Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . 59
    4.2.1.1 Proposed technique . . . . . . . . . . . . . . . . . 60
  4.2.2 Results and evaluation . . . . . . . . . . . . . . . . . . . . . 62
    4.2.2.1 All combinations evaluation . . . . . . . . . . . . 62
    4.2.2.2 Preprocessing methods effect . . . . . . . . . . . . 63
    4.2.2.3 The optimal system . . . . . . . . . . . . . . . . . 63
4.3 Chapter summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64

Chapter 5

K-nearest neighbor algorithm for online signature verification 66
  5.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
  5.2 Related work . . . . . . . . . . . . . . . . . . . . . . . . . . . . 68
  5.3 Evaluation of the algorithm parameters’ effect . . . . . . . . . . . 68
    5.3.1 Number of references and threshold scale . . . . . . . . 69
    5.3.2 j and k values . . . . . . . . . . . . . . . . . . . . . . 70
    5.3.3 jk-NN performance . . . . . . . . . . . . . . . . . . . . 71
  5.4 Optimized jk-NN based online signature verification . . . . . . . 73
  5.5 k-NN and DTW for mobile scenario online signature verification . . 74
    5.5.1 Methodology . . . . . . . . . . . . . . . . . . . . . . . . . 74
    5.5.2 SVC2021 & SVC-onGoing competitions . . . . . . . . . 75
    5.5.3 Experimental results . . . . . . . . . . . . . . . . . . . . 76
      5.5.3.1 DeepSignDB . . . . . . . . . . . . . . . . . . . . . 76
      5.5.3.2 SVC2021_EvalDB . . . . . . . . . . . . . . . . . 77
  5.6 Chapter summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
List of Figures

1.1 Online vs offline signatures [Bibi et al., 2020] .................................. 2

2.1 Main steps of the verification phase. ............................................... 9
2.2 Wacom Intus 3 [Venkittarayan et al., 2010] ...................................... 10
2.3 DeepSignDB database structure. .................................................. 14

3.1 Available combinations in the system. ............................................ 33
3.2 Effect of training set selection strategy on the average EER. .......... 35
3.3 Effects of rotation normalization on the average EER. .................. 41
3.4 Average EER for all remaining configurations grouped according to datasets. ................................................................. 42
3.5 Summary. .................................................................................. 44

4.1 Combinations of the applied verifiers. ........................................... 53
4.2 Expected (left) vs. typical (right) behavior of the EER as a function of sample points on SVC2004. ................................................. 55
4.3 Examples from other databases of the effect of sampling frequency on the EER. ................................................................. 56
4.4 Sampling frequencies of the best results for all the tests. ............. 57
4.5 Sample counts of the best results for all the tests. ......................... 57
4.6 The combinations of the tests applied in the experiments. .......... 61
4.7 Proposed method. ................................................................. 62
4.8 Percentage of experiments (vertical axis) where the accuracy improved when using the signer-dependant sampling frequency approach (all the tests). ............................................. 63
4.9 Percentage of experiments (vertical axis) where the accuracy improved when using the signer-dependant sampling frequency approach (using z-normalization). ..................... 64
4.10 Percentage of experiments (vertical axis) where the accuracy improved when using the signer-dependant sampling frequency approach (using z-normalization and 6 samples).......................... 64

5.1 $k$-NN (Left) and $jk$-NN (Right)........................................... 67
5.2 Accuracy improvement of the $jk$-NN algorithm comparing to the $k$-NN algorithm................................................................. 71

6.1 Dependency graph of main namespaces........................................ 82
6.2 Relevant loader classes............................................................... 82
List of Tables

2.1 Common public online signature databases. .................................. 16
2.2 Preprocessing methods used in the last two decades .................... 18
2.3 Feature used in the verification system ...................................... 20
2.4 Verification techniques used in the last two decades .................... 22
2.5 A Comparison between methods presented in the last few years, S=stylus, F=finger. ........................................................... 24

3.1 Percentage of all Configurations, Where Removing the Pen-up Strokes Resulted in a Lower EER .................................................. 37
3.2 Percentage of all Configurations, Where Inserting Synthetic Pen-up Strokes Resulted in a Lower EER .................................................. 38
3.3 Comparison of Euclidean and Manhattan Distance Based Verifier Configurations ................................................................. 39
3.4 Average EER of Configurations Using the Given Translation and Scaling Methods for Preprocessing .................. 40
3.5 Comparison of Three Normalization Approaches for the Remaining Configuration Triplets ........................................................... 41
3.6 Comparison of Results on SVC2004 ............................................ 43
3.7 Comparison of Results on MCYT100 ........................................... 44
3.8 Comparison of Results on the SigComp’11 and SigWiComp2015 Databases ................................................................. 45

4.1 Best sampling frequencies, sample counts, and EER for each combination for each database. .................................................. 54
4.2 Databases used and their statistics .............................................. 55
4.3 A comparison with recent similar results in the field .................... 59

5.1 The best results of the experiments. ............................................ 70
5.2 The effect of $j$ (up to down) and $k$ (left to right) on FAR (up) and FRR (bottom) .................................................. 72
5.3 Min FAR and AER of the proposed algorithm ..................... 74
5.4 SVC2021-DeepSignDB - Task2 results ............................. 77
5.5 SVC2021- SVC2021_EvalDB - Task2 results ....................... 77
5.6 SigStat team results .................................................. 78
A.1 Summary of related own publications ............................. 87
A.2 The error rate for each database .................................. 92
A.3 The error rate for each database .................................. 94
A.4 The AER for each database ........................................ 96
Biometrics are representations of the body and measurements related to human features. Biometric identification and authentication are used to identify an individual on the basis of its distinctive features and access control in computer science. They are challenging tasks when it comes to an automated system with high accuracy.

Biometrics are divided into two groups, physiological and behavioral. Physiological biometrics are acquired naturally by birth, such as hand geometry, finger-print, eye-print, DNA, and face shape [Bolle et al., 2013] [Impedovo and Pirlo, 2008]. While behavioral biometrics refer to some actions that are carried out and developed over time and might be related to the place of birth or education [Faundez-Zanuy and Monte-Moreno, 2005] such as signatures, voice, typing patterns, and physical movements [Veeramachaneni et al., 2005]. One sample of a physiological biometric can provide enough information about the person to be recognized or authorized, making it more reliable compared to behavioral biometrics, which can be affected by several factors that provide different information each time.

Various behavioral biometrics exist, handwriting signatures occupy an exceptional place in this field [Fairhurst, 1997, Fairhurst and Kaplani, 2003, Plamondon and Lorette, 1989a]. The term signature is generally means the signing with one’s hand. A signature aims to authenticate a written document or
provide notice of its source and oblige the person who signs a written document to comply with the document’s requirements.

Signatures are the most widely accepted biometric trait by law for verification purposes [Jain et al., 2020]. In computer-aided signature verification, there are two types of signatures distinguished by the input method. The signature is captured using regular pen and paper in offline or static signatures, then scanned as a file image. While in online or dynamic signatures, the whole signing process is acquired and recorded using special digital devices, e.g., a tablet or digital pens. These devices can capture several features like position and pressure as a function of time, thereby adding valuable extra information for the verification process (see Figure 1.1).

![Figure 1.1: Online vs offline signatures [Bibi et al., 2020]](image)

There is one thing in common for most signature authentication systems. They work primarily like black boxes (mainly because of the software’s AI-based approach), providing only little helpful information on the reasons for a decision [Kovari and Charaf, 2013]. This makes their enhancement a difficult challenge, and also make it difficult to introduce in forensic examinations.

In this work, I focused on the online signature verification systems. I investigated and evaluated the effect of different algorithms used in various steps of
the verification process. I also studied the effect of the sampling frequency and signature points count on the system’s accuracy. Also, some classification methods were evaluated and analyzed. Based on the mentioned studies, I proposed several verification systems that enhanced the verification results’ accuracy.

1.1 Thesis contributions

As long as human experts can outperform automatic signature verifiers, there is a need for improvement. Therefore, the research objective was to establish a competitive signature verification system and evaluation methodologies suitable for the classification of signatures. The work started by analyzing and summarizing the previous work and results in the field. These results were concluded, evaluated, and compared in a way that will hopefully help the researchers in the field. Further, the existing algorithms and methods were combined to form many verification scenarios to analyze and evaluate the most useful ones and form a more robust verification system that provides better results. This is followed by a novel method of using the sampling frequency as an essential factor in the verification, leading to a new competitive verification system. Other classification methods were also improved and optimized to form a verification system with more accurate results.

These results were analyzed and published in several journals and conferences. Additionally, the core project used in this thesis is made publicly available for the research community, as an open source project, along with all the relevant algorithms introduced in this work for anyone who is interested in the world of signature verification.

Taking these aspects into account, my goals were the following:

- Review of most relevant journal and conference papers from the last 30 years in the field of signature verification.

- Conduct an extensive survey of possible preprocessing and verification methods.

- Examine and – when necessary – improve existing methods used in each step of the verification process.
Chapter 1. Introduction

- Apply methods to signature databases and prove the feasibility of their practical application.

- Apply a large-scale evaluation of signature verification steps.

- Present a new verification system using the analysis of the most effective methods used in each verification step.

- Build a new verification system using the sampling frequency effect on the verification accuracy.

- Present a verification system using an enhanced and optimized classification method.

1.2 Thesis structure

After the introduction in the first chapter, the other chapters of the thesis are organized as follows:

- **Chapter 2**: epitomizes the motivations of the accomplished results of this thesis. Furthermore, discusses the main steps of the online signature verification systems and presents each step’s main algorithms. It also shows the most important approaches and verification results in the field and compares them in a scientific style. The common available databases are also presented and compared.

- **Chapter 3**: analyzes and investigates the main approaches of each verification step and their combinations, which may define numerous significant features of a signature verification process and estimate their effect on the accuracy of the classification step. These results were used to present optimal verification systems with competitive results compared to the literature review.

- **Chapter 4**: investigates the effect of the sampling frequency and the number of signature points on the accuracy of the verification accuracy. Moreover,
it presents a novel verification system using signer-dependent verification system with high and competitive accuracy.

- **Chapter 5:** evaluate the effect of using $k$-nearest neighbor for signature classification and its parameters. Furthermore, it presents an improved and optimized $jk$-nearest neighbour method that improves the verification system’s accuracy compared to the standard method and a new verification system using a combination of both $k$-nearest neighbor and dynamic time warping algorithm.

- **Chapter 6:** discusses the applications of the practical relevance of the research, mainly our SigStat project that was developed over the past few years and is publicly available to the researchers in the field as a benchmark for both online and offline signature verification systems. And presents a summary and evaluation of the work.

- **Appendix A:** presents a summary of the main results and conclusions.
Chapter 2

Background and related work

In this chapter, the work’s motivation and the main steps of the online signature verification systems are discussed. The most important approaches and verification results and the common available databases in the field are shown and compared.

2.1 Terms and concepts

This section defines the main terms and concepts used in the online signature verification field to avoid any ambiguous interpretations through the thesis.

- **Signature**: A person’s name, nickname or symbol drawn by the person used for identification and authorization in documents.

- **Online signature**: is a signature acquired using special electronic devices that has the ability to capture several features as functions of time.

- **Genuine signature**: is a personal name or symbol drawn and obtained by that specific person.

- **Forged signature**: is a signature that imitates the genuine signature of a specific signer but was not created by him/her. Forgeries can be divided into several sub-classes.
  - **Random forgery**: is a random signature that is obtained by someone who doesn’t know the original shape of the genuine signature.
- Skilled forgery: is a forged signature created by a skillful forger after a thorough study of the original signatures.

- **Reference signature**: is a signature provided by the original signer used for training a verification system.

- **Sample signature (questioned signature)**: is a signature acquired from a signer who is anonymous to the verification system, used to test a verification system.

- **Accepted sample**: a sample, which was classified a genuine signature by the verification system.

- **Rejected sample**: a sample classified as a forgery by the verification system.

- **Signature verification system (SVS)**: is a verification system used to classify questioned signatures as genuine or forged.

### 2.2 Performance metrics

The performance evaluation of any online signature verification system is measured in terms of error rates [Neyman and Pearson, 1933, Tipton and Krause, 2006]. Two types of errors may occur during this process. The first one occurs when a genuine signature is classified as forged (false rejection rate (\(FRR\))), and the second one occurs when a forged signature is classified as genuine (false acceptance rate (\(FAR\))). In the testing phase, some genuine signatures (different from the signatures used as references) are used to calculate the FRR as follows:

\[
FRR = \frac{\text{Genuine signatures classified as forged in the testing set}}{\text{Genuine signatures in the testing set}} \tag{2.1}
\]

while some forged signatures are used to calculate the FAR as follows:

\[
FAR = \frac{\text{Forged signatures classified as genuine in the testing set}}{\text{Forged signatures in the testing set}} \tag{2.2}
\]
These two types of errors are taken into consideration while evaluating the verification system. A trade-off between the two errors types must be taken into account to evaluate signature verification systems. If the system is very strict, the FRR will increase and the FAR will decrease. Conversely, the FAR increases and the FRR decreases when the system is relaxed.

The equal error rate (EER) is the point where FRR and FAR are equal, where the system can be tuned to this point. The EER is widely accepted for online signature verification performance evaluation. In some cases, it is not possible to find EER where FAR=FRR, thus EER can also represent a minimal difference between the two values, or the first value obtained after the value of one exceeds the other.

While other researchers used average error rate (AER) by calculating the average of the two types:

\[
AER = \frac{FRR + FAR}{2}
\]  

(2.3)

There is no one criteria or method of system performance evaluation that fit all situations and real-life scenarios. In most cases, the application would be biased toward one of the two error types, mainly to keep one type as lower as possible.

In addition, it is worth mentioning that AER and EER do not provide enough information about the system performance when the system is applied under stricter rules or different scenarios. However, they provide a good indication about how the system might perform compared to other systems within the similar circumstances.

2.3 Background

The first use of the signature verification concept goes back to the 439 AD where it was used for document authentication in the Roman Empire [Bibi et al., 2020]. However, the first automatic signature verification system started developing from the 1960s [Fauziyah et al., 2009]. The first automated signature recognition system was presented in 1965 by North America Aviation. The first online and offline verification system was presented in 1973 [Nagel and Rosenfeld, 1973]. After that, thousands of research papers have been published in the field. In 2004,
the first signature verification competition was held at the Hong Kong University [Yeung et al., 2004].

2.4 Verification process and related work

Any signature verification system typically consists of four main steps (Figure 2.1). The first step is data acquisition, where the signatures are acquired using dedicated device (typically digitizing tablet).

The next step is preprocessing. Even when the signer provides the signatures under similar circumstances, there will always be some differences in size or location that may hinder their comparison. Thus, in the preprocessing step, methods such as scaling [Jindal et al., 2018], alignment [Ahrabian and BabaAli, 2019] [Xia et al., 2017], rotation [Mohammadi and Faez, 2012], or z-normalization [Auckenthaler et al., 2000] can be applied to enhance the similarity measurement in the later steps. After that, feature extraction is applied, where several features can be extracted, such as the position, speed, pressure, and azimuth.

In the verification step, some verification and similarity measurement methods are applied to decide whether a signature is genuine or forged. There are several approaches used for this purpose, such as dynamic time warping (DTW), neural networks [Lai et al., 2017], or hidden Markov models (HMMs) [Farimani and Jahan, 2018]. Among these methods, DTW has shown the most promising results [Malik et al., 2015].

This section will discuss each main step of the online signature verification process with the common algorithms and methods applied. The previous results of each method will also be shown, and the state of the art will be presented and compared.

![Figure 2.1: Main steps of the verification phase.](image-url)


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text


text
2.4.2 Databases

Signature databases are essential for developing and testing any signature verification system. A signature database is a collection of signatures that are obtained from different signers and then stored as images or digitized files. Some of these databases are publicly available for the researchers. They contain different sets, which in some databases are divided into training and evaluation sets folders.

Signature databases might vary in some features, such as the number of signees. Increasing the number of signees will provide diversity for the database. To ensure the diversity of the database, signees shouldn’t be chosen from a similar group, for example, same age or profession. The number of signees in the available online signature databases is between 20–1526. The number of signatures is also important, particularly the number of genuine signatures since higher number of genuine signatures ensures enough data to train the system and enough testing data. The quality of the forged signatures also plays an important role in the system evaluation. Skilled forgeries make it harder to distinguish from genuine signatures, unlike random forgeries. The number of forgeries should be in the same range as the genuine signatures. In this section, I introduce and compare the most commonly used databases.

2.4.2.1 SVC2004

The SVC2004 (Signature verification competition 2004) [Yeung et al., 2004] is the oldest available online signature database, yet still one of the most-commonly used databases in the field. The data were firstly used in the signature verification competition, which aimed to compare different verification methods systematically. It contains two different databases with the same signatures. While one of them contains X and Y coordinates, timestamp, and pen status, the other one contains more additional features such as azimuth, altitude, and pressure [Rashidi et al., 2012]. Each database contains signatures from 40 signers. Each signer has 20 genuine signatures and 20 skilled forgeries.
2.4.2.2 MCYT

The MCYT (Ministerio de Ciencia y Tecnología, Spanish Ministry of Science and Technology) is a large-scale population database. This database has high number of enrolled signees and different modalities per signee with different number of samples. It contains signatures from 330 individuals; each signer has 25 genuine signatures and 25 forged signatures collected for him/her [Yanikoglu and Kholmatov, 2009]. The skilled forgeries are collected by five forgers. The signature features consist of position, pressure, azimuth, and altitude. The tablet used here has a sampling rate of 100 Hz [Ortega-Garcia et al., 2003].

2.4.2.3 SigComp’11

The signature verification competition 2011 consists of several sub-databases, which were introduced by Netherlands Forensic Institute for Forensic Science and the German Research Center for Artificial Intelligence. One of these sub-databases is the online Dutch database, which consists of 1907 signatures provided by 64 signers. The signatures were collected using WACOM Intuos 3 and MovAlyzer software with a sampling rate of 200 Hz and a resolution of 2000 line/cm.

SigComp’11 (Chinese) is another sub-database of the signature verification 2011 database and consists of 1339 signatures. The number of signers is 20, and the acquisition device is WACOM Intuos3.

2.4.2.4 SigWiComp2015 German

Signature competition 2015 database consists of sub-databases with several languages like German, Bengali, and Italian. The SigWiComp German database was collected using Ando pen instead of a tablet. The database features are horizontal position, vertical position, and pressure. It was collected with a 75 Hz frequency from 30 signers with total signatures of 750.

2.4.2.5 SUSIG

The SUSIG (Sabanci University Signature database) database is also widely used in this regard. It consists of two parts: the visual sub-corpus part collected using
Interlink ePad-ink tablet, and the blind sub-corpus part, which was collected using Wacom Graphire 2 tablet. In the first part, each signer has 20 genuine and 10 forged signatures. The second part consists of 10 genuine and 10 forged samples by each signer [Rashidi et al., 2012]. The Interlink tablet has an LCD screen dimensions of $3 \times 2.20$ inches, a resolution of 300 dpi, and its sampling rate is 100 Hz. The Wacom tablet’s dimensions are $5.02 \times 3.65$ inches with a 100 Hz sampling rate [Khalil et al., 2009] [Kholmatov and Yanikoglu, 2009].

2.4.2.6 BioSecure

BioSecure Multimodal Database was captured using WACOM Intuos 3 pen tablet with a sampling rate of 100 Hz. The database was captured in two sessions, and the duration between the first and second sessions was two months. For each of the 120 signers, 30 genuine and 20 forgeries were collected. The features captured in this database are $X$ and $Y$ coordinates, azimuth angle, altitude angle, time stamp, and pressure [Tolosana et al., 2015a].

2.4.2.7 SigWiComp2013

A Japanese signature data-set was used in SigWiComp2013 [Malik et al., 2013]. It has been captured with an HP EliteBook 2730p tablet and used for the database mentioned above, with a sampling rate of 200 Hz and 50 dpi. The number of the users was 31; for each user, 42 genuine signatures and 36 forged signatures were collected [Zeinali and BabaAli, 2017][Ahrabian and BabaAli, 2019].

2.4.2.8 SIGMA

SIGMA is a Malaysian signature database. It consists of over 6,390 genuine signatures and 2,130 forged signatures for Malaysian nationals [Ahmad et al., 2008]. All the forged signatures in the SIGMA database are skilled forgeries. Wacom Intuos 3 digitizing tablet has been used for signature acquisition. The tablet’s resolution is 5080 dpi, and it has a sampling frequency of 200 Hz and 1024 pressure levels. This database was used in [Malallah et al., 2015] to test the signature verification system.
2.4.2.9 DeepSignDB

DeepSignDB database is a combination of several databases. It contains two types of input signatures, stylus and finger inputs. The number of signees is 1526, providing more than 70,000 signatures from both types using eight different capture devices [Tolosana et al., 2021a]. It contains both random and skilled forged signatures. The structure of the DeepSignDB is show in Figure 2.3.

![DeepSignDB Database Structure](image)

Figure 2.3: DeepSignDB database structure.

2.4.2.10 SVC2021_EvalDB

The SVC2021_EvalDB was acquired for the SVC2021 online signature verification competition. It has both office and mobile scenarios input signatures. The office scenario signatures were acquired from 75 signers using the Wacom STU-530 device in two separate sessions with a one-week time gap. The X and Y spatial
coordinates, pressure, timestamp, and pen-up trajectories were recorded for these signatures. The mobile scenario signatures were acquired by a total of 119 signers in four and six separated sessions with at least three weeks gap between the first and last sessions. Only the X and Y spatial coordinates, and timestamps were recorded. In both scenarios, for each signer, there are 8 genuine and 16 forged signatures [Tolosana et al., 2021b].

2.4.2.11 ATVs

The common databases such as MCYT, SVC2004, and SUSIG are very important because they are human-made signatures. However, at the same time, they contain a limited amount of data and also store private data. To avoid these restrictions, synthetic databases such as ATVs were produced using a generative model based on information obtained after analyzing genuine signatures. These databases are not restricted to the limitations mentioned before. The disadvantages are that the signatures are not human-made and do not represent real situations. ATVs follows the pattern of western signatures (left to right signatures) [Galbally et al., 2012].

2.4.2.12 Own databases

Some other researchers have used their own databases of signatures in their work. Ibrahim et al. [Ibrahim et al., 2010] used their own database, which consisted of signatures from 25 signers; the total number of signatures was 21,250 (15,000 genuine and 6,250 forgeries).

Table 2.1 shows a comparison between the mentioned databases.

2.4.3 Preprocessing

The aim of preprocessing is to enhance the captured signatures in order to obtain the same type of feature information, like position or velocity, and thereby improve the system’s accuracy. In this section, we discuss the most common preprocessing methods. The most common preprocessing step is normalization. We can differentiate between several of its kinds: normalization of the horizontal position and vertical position (alignment), minimum and maximum normaliza-
Table 2.1: Common public online signature databases.

<table>
<thead>
<tr>
<th>Database</th>
<th>Signers</th>
<th>Genuine</th>
<th>Skilled</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC2004</td>
<td>40</td>
<td>800</td>
<td>800</td>
<td>100 Hz</td>
</tr>
<tr>
<td>MCYT</td>
<td>330</td>
<td>6600</td>
<td>8250</td>
<td>100 Hz</td>
</tr>
<tr>
<td>SigComp’11 Dutch</td>
<td>64</td>
<td>1536</td>
<td>~820</td>
<td>200 Hz</td>
</tr>
<tr>
<td>SigComp’11 Chinese</td>
<td>20</td>
<td>~400</td>
<td>~940</td>
<td>200 Hz</td>
</tr>
<tr>
<td>SigWiComp’15</td>
<td>30</td>
<td>450</td>
<td>300</td>
<td>75 Hz</td>
</tr>
<tr>
<td>BioSecure</td>
<td>120</td>
<td>3600</td>
<td>2400</td>
<td>100 Hz</td>
</tr>
<tr>
<td>SigWiComp’13</td>
<td>31</td>
<td>1302</td>
<td>1116</td>
<td>200 Hz</td>
</tr>
<tr>
<td>SUSIG</td>
<td>94</td>
<td>1880</td>
<td>940</td>
<td>100 Hz</td>
</tr>
<tr>
<td>SIGMA</td>
<td>200</td>
<td>2000</td>
<td>1000</td>
<td>200 Hz</td>
</tr>
<tr>
<td>DeepSignDB</td>
<td>1526</td>
<td>~70000</td>
<td>Multiple Freqs</td>
<td></td>
</tr>
<tr>
<td>SVC2021_EvalDB</td>
<td>194</td>
<td>1552</td>
<td>3104</td>
<td>Multiple Freqs</td>
</tr>
<tr>
<td>ATVS</td>
<td>350</td>
<td>8750</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

tion [Ansari et al., 2013], length normalization (scaling) [Al-Shoshan, 2006], time normalization [López-García et al., 2013] and z-normalization [Diaz et al., 2015]. Moreover, up-sampling normalization and down-sampling normalization can be applied as a preprocessing step [Malallah et al., 2015]. [Al-Shoshan, 2006] Normalized moments and normalized Fourier descriptors were used to make the signature features such as scale, rotation, and displacement invariant. Signature alignment is widely used for preprocessing. It is done by subtracting a value from each point of the signature to re-align it so that it starts or ends at a specific point, which improves the results of the similarity matching of the signatures. Some articles refer to alignment as position normalization, location normalization, or translation. Xia et al. [Xia et al., 2018] align the signature based on Gaussian Mixture Model to obtain the best matching results. Some other variants for alignment were also introduced in some research’s [Ibrahim et al., 2010][Kholmatov and Yanikoglu, 2005].

Signature scaling (or length normalization) is also used to reduce the error in similarity measurement by scaling the signature by a specific ratio. Scaling can be applied horizontally, vertically, or in both directions. Similar to alignment, this is a common approach in signature verification systems [Nilchiyan et al., 2015][Chadha et al., 2011]. A signer may sign his/her signature with different rotation angles. In order to obtain better results, rotation preprocessing may be applied to eliminate these differences [Xia et al., 2017][Wirotius et al., 2004]. Rotation based on orthogonal regression
was used in [Fayyaz et al., 2015b] and the rotation angle was removed using DTW in [Mohammadi and Faez, 2012].

Pen-up duration refers to the time when the pen is not touching the input device. Forgery signatures usually take a longer time in the pen-up duration compared to the original signatures. This can be used to detect the forged signatures [Yanikoglu and Kholmatov, 2009]. In most cases, this feature is discarded. Yanikoglu et al. [Yanikoglu and Kholmatov, 2009] filled the pen-up duration with imaginary points using various methods (as an example, one can fill up imaginary points every 30 ms while using a 100 Hz input device). In the same research, Yanikoglu et al. applied drift and mean removal to remove the signature’s baseline drift component. Linear regression with the least square error was used for this purpose, while the mean removal was done by subtracting the mean value of the points.

Some points of the signature may have falsely become part of it because of the high sensitivity of the input device. According to [Ibrahim et al., 2010] these points have low pressure values and should be set to zero since they are not part of the signature. Thus, a threshold can be applied to remove these false captured values. Ibrahim et al. [Ibrahim et al., 2010] used Equation (2.5) to calculate this threshold:

\[
    t^j_i = \frac{1}{M} \sum_{m=1}^{M} z^j_i (m) - \sqrt{\frac{1}{M} \sum_{m=1}^{M} \left(z^j_i (m) - \frac{1}{M} \sum_{m=1}^{M} z^j_i (m)\right)^2} \tag{2.5}
\]

Some other preprocessing steps are used to enhance the results of the signature verification. In [López-García et al., 2013], the average signals are transformed to an equally-spaced 256-point using linear interpolation. Arora et al. [Arora et al., 2015] converted the signature into a coordinate system by calculating the derivatives of various parameters up to the second order. This produced a vector of the first and second parameters of the coordinates, the pressure, the speed, and the angle, which were used in their verification system [see Equation 2.6].

\[
    V_i = [X^1, Y^1, P^1, \phi^1, X^2, Y^2, P^2, \phi^2] \tag{2.6}
\]
Nilchiyan et al. [Nilchiyan et al., 2015] used the method of resampling using cubic spline interpolation. The number of signature points is not the same for the signature and the reference signature; thus, resampling is applied to make the number of sampled points the same as the reference points. Table 2.2 shows a summary of the mentioned preprocessing methods and the articles in which they are used.

Table 2.2: Preprocessing methods used in the last two decades

<table>
<thead>
<tr>
<th>Preprocessing</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalization</td>
<td>[Yanikoglu and Kholmatov, 2009] [Tolosana et al., 2015a]</td>
</tr>
<tr>
<td></td>
<td>[Ahrabian and BabaAli, 2019] [Ibrahim et al., 2010]</td>
</tr>
<tr>
<td></td>
<td>[Xia et al., 2017] [Kholmatov and Yanikoglu, 2005]</td>
</tr>
<tr>
<td></td>
<td>[Radhika and Gopika, 2015] [Wirotius et al., 2004]</td>
</tr>
<tr>
<td></td>
<td>[Ibrahim et al., 2010] [Mohammadi and Faez, 2012]</td>
</tr>
<tr>
<td></td>
<td>[Nilchiyan et al., 2015] [López-García et al., 2013]</td>
</tr>
<tr>
<td></td>
<td>[Khan et al., 2006] [Chadha et al., 2011] [Jindal et al., 2018]</td>
</tr>
<tr>
<td></td>
<td>[Chadha et al., 2011] [Wirotius et al., 2004]</td>
</tr>
<tr>
<td></td>
<td>[Fayyaz et al., 2015b] [Jindal et al., 2018]</td>
</tr>
<tr>
<td></td>
<td>[López-García et al., 2013] [Díaz et al., 2015]</td>
</tr>
<tr>
<td></td>
<td>[Pascual-Gaspar et al., 2011]</td>
</tr>
<tr>
<td>Down-sampling normalization</td>
<td>[Malallah et al., 2015] [Ansari et al., 2013]</td>
</tr>
<tr>
<td>Min. normalization</td>
<td>[Ansari et al., 2013]</td>
</tr>
<tr>
<td>Max. normalization</td>
<td>[Ansari et al., 2013]</td>
</tr>
<tr>
<td>Pen-up duration</td>
<td>[Yanikoglu and Kholmatov, 2009]</td>
</tr>
<tr>
<td>Zero pressure removal</td>
<td>[Ibrahim et al., 2010]</td>
</tr>
<tr>
<td>Smoothing</td>
<td>[Ibrahim et al., 2010] [Radhika and Gopika, 2015]</td>
</tr>
<tr>
<td></td>
<td>[Jain et al., 2002] [Wirotius et al., 2004]</td>
</tr>
<tr>
<td>Equal spacing</td>
<td>[López-García et al., 2013] [Francis et al., 2015]</td>
</tr>
<tr>
<td></td>
<td>[Jain et al., 2002]</td>
</tr>
<tr>
<td>Resampling using cubic spline</td>
<td>[Ibrahim et al., 2010] [Nilchiyan et al., 2015]</td>
</tr>
<tr>
<td></td>
<td>[Radhika and Gopika, 2015]</td>
</tr>
<tr>
<td>Up-sampling normalization</td>
<td>[Malallah et al., 2015]</td>
</tr>
<tr>
<td>Down-sampling normalization</td>
<td>[Malallah et al., 2015]</td>
</tr>
<tr>
<td>Drift removal</td>
<td>[Yanikoglu and Kholmatov, 2009]</td>
</tr>
<tr>
<td>Filtering</td>
<td>[Jindal et al., 2018]</td>
</tr>
<tr>
<td>Noise removal</td>
<td>[Jindal et al., 2018]</td>
</tr>
<tr>
<td>Filtering</td>
<td>[López-García et al., 2013]</td>
</tr>
</tbody>
</table>
2.4.4 Feature extraction

Feature extraction is one of the main steps of signature verification. There are two types of features: function and parameter features. In the case of function features, the values are described as a function of time, while in the parameter features, they are considered a vector of elements. Usually, the function feature consumes more time for processing than the parameter features [Plamondon and Lorette, 1989b], but they give better results.

The parameter features are divided into two types, local and global. Global features describe the whole signature, such as number of pen lifts and time duration, while the local features are extracted from specific parts of signatures.

The most widely-used features are position, velocity, and acceleration. Position can be acquired directly from the input device or pen movement, while some features can be derived numerically from other features. However, recently, specific devices have been developed to directly provide these features during the signing process [Ohishi et al., 2001, Omata et al., 2001].

The other type of features is the parameter features, which includes total signature duration, number of pen lifts while signing, and pen-up/pen-down time ratio. Other parameters like average, max, and min of position, speed, acceleration, etc., can be derived from the analysis of direction or other features. Fourier transform and wavelet transform can derive the coefficient used in signature verification. Table 2.3 shows some of the features used in the verification system.

2.4.5 Similarity Measurements and Classification

After the signature acquisition, preprocessing, and feature extraction, the signatures are ready to be compared with the original signatures. One of the most popular methods for distance-based measurement is dynamic time warping (DTW). DTW algorithm finds the best non-linear alignment of two-time series so that the overall distance between them is minimized. It has shown promising results in the signature verification field [Malik et al., 2015]. As it calculates the distance between two signatures, it is commonly used in threshold-based classification approaches, like in [Song et al., 2016, Fischer et al., 2015].
Table 2.3: Feature used in the verification system

<table>
<thead>
<tr>
<th>Feature</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>[Zeinali and BabaAli, 2017] [Ibrahim et al., 2010] [Liu et al., 2014] [Xia et al., 2017] [Pascual-Gaspar et al., 2011] [Fischer et al., 2015]</td>
</tr>
<tr>
<td>Velocity</td>
<td>[Zeinali and BabaAli, 2017] [Ibrahim et al., 2010] [Pascual-Gaspar et al., 2011] [Fischer et al., 2015]</td>
</tr>
<tr>
<td>Pressure</td>
<td>[Ibrahim et al., 2010] [Pascual-Gaspar et al., 2011] [Liu et al., 2014] [Sharma and Sundaram, 2016c]</td>
</tr>
<tr>
<td>Acceleration</td>
<td>Ye et al., 2017 [Fischer et al., 2015]</td>
</tr>
<tr>
<td>Wavelet-based features</td>
<td>Nilchiyan et al., 2015</td>
</tr>
<tr>
<td>Azimuth</td>
<td>Liu et al., 2014</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Liu et al., 2014</td>
</tr>
<tr>
<td>Normalized Fourier Descriptor</td>
<td>Al-Shoshan, 2006</td>
</tr>
<tr>
<td>Normalized central moments</td>
<td>Al-Shoshan, 2006</td>
</tr>
<tr>
<td>Angle based features</td>
<td>Sharma and Sundaram, 2016c</td>
</tr>
<tr>
<td>Log curvature radius</td>
<td>Zeinali and BabaAli, 2017</td>
</tr>
<tr>
<td>Total acceleration magnitude</td>
<td>Zeinali and BabaAli, 2017</td>
</tr>
<tr>
<td>DCT coefficients</td>
<td>Rashidi et al., 2012</td>
</tr>
</tbody>
</table>

Yanikoglu et al. [Yanikoglu and Kholmatov, 2009] presented a novel online signature verification system based on the Fast Fourier Transform (FFT); this approach enhanced the performance of the DTW by up to about 25%. Another approach presented in [Sharma and Sundaram, 2016c] enhanced DTW by utilizing the code-vectors created through a Vector-Quantization (VQ) method. The combination of scoring/voting methods of DTW/VQ showed an enhanced result of the signature verification. VQ approaches were also presented in [Pascual-Gaspar et al., 2011], which enhanced the verification system and increased the speed of the process. Xia et al. presented a modified DTW with a signature curse in order to improve the system’s efficiency [Xia et al., 2017]. Besides DTW, other different distance-based approaches are used in signature verification: Mahalanobis distance [Tolosana et al., 2015a], Kolmogorov Smirnov distance [Griechisch et al., 2014], and Manhattan, Chebyshev, and Euclidean distances [Ibrahim et al., 2010, Arora et al., 2015].

Another verification technique using discrete cosine transform (DCT) and sparse representation was presented in [Liu et al., 2014], and it showed a com-
petitive result compared to functional approaches. Moreover, it is simpler and easier for the matching process.

Signature verification systems have not used plenty of deep learning methods since these second-mentioned require a large amount of data for training, which is not publicly available for signatures, as for other biometrics such as voice and face [Kemelmacher-Shlizerman et al., 2016]. In [Al-Shoshan, 2006], a multi-layer perceptron (MLP) neural network was used for the classification with one input layer, one hidden layer, and one output layer. MLP was also presented in [Hefny and Moustafa, 2019], it was applied on the SigComp20'11(Dutch) database, where they use Legendre polynomials coefficients as features. A novel back-propagating recurrent neural network (RNN) was presented to improve the verification performance in [Lai and Jin, 2018]. Convolutional Neural Networks (CNNs) method was also used in few papers. In [Wu et al., 2019a], a combination of CNNs and DTW was used and applied on the MCYT database. One dimensional CNN verification system was used in [Lai et al., 2020]. Tolosana et al. proposed an online signature verification system using Time-Aligned Recurrent Neural Net-works (TA-RNN), which combines DTW and RNNs. It was applied on the recently published DeepSignDB database [Tolosana et al., 2021]. Some other RNNs systems were proposed using Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) [Tolosana et al., 2018], or Bidirectional Recurrent Neural Network [Nathwani, 2020].

Length-normalized path signatures (LNPS) were also applied to the system because of its properties, such as scale and rotation invariance. Support vector machines (SVMs) contain the supervised model learning algorithms used in classification [Cortes, 1995]. The advantages of using SVM are that they have a convex objective function with efficient training algorithms and good generalization properties [El-Henawy et al., 2013].

Many other approaches can be used for the verification step, such as Gaussian Mixture Model (GMM) [Xia et al., 2017, López-García et al., 2013], Hidden Markov Model (HMM) [Tolosana et al., 2015c], K-Fold Cross-Validation [Nilchiyan et al., 2015], Parzen Window Classifier [Rashidi et al., 2012], Naïve Bayesian (NB), nearest neighbor, principal component analysis, and linear discrim-
Table 2.4: Verification techniques used in the last two decades

<table>
<thead>
<tr>
<th>Method</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance-based approaches:</td>
<td></td>
</tr>
<tr>
<td>DTW-based approaches</td>
<td></td>
</tr>
<tr>
<td>DTW/threshold</td>
<td>Khalil et al., 2009</td>
</tr>
<tr>
<td></td>
<td>Song et al., 2016</td>
</tr>
<tr>
<td></td>
<td>Mohammadi and Faez, 2012</td>
</tr>
<tr>
<td></td>
<td>Fischer et al., 2015</td>
</tr>
<tr>
<td></td>
<td>Tolosana et al., 2015b</td>
</tr>
<tr>
<td></td>
<td>Diaz et al., 2015</td>
</tr>
<tr>
<td>DTW/FFT</td>
<td>Yanikoglu and Kholmatov, 2009</td>
</tr>
<tr>
<td>DTW/Vector-Quantization</td>
<td>Sharma and Sundaram, 2016c</td>
</tr>
<tr>
<td>DTW/curve constraint and GMM for matching</td>
<td>Xia et al., 2017</td>
</tr>
<tr>
<td>Other distance-based approaches</td>
<td></td>
</tr>
<tr>
<td>Mahalanobis distance</td>
<td>Malik et al., 2013</td>
</tr>
<tr>
<td>Manhattan distance</td>
<td>Arora et al., 2015</td>
</tr>
<tr>
<td>Chebyshev distance</td>
<td>Arora et al., 2015</td>
</tr>
<tr>
<td>Euclidean distance</td>
<td>Ibrahim et al., 2010</td>
</tr>
<tr>
<td></td>
<td>Arora et al., 2015</td>
</tr>
<tr>
<td>Vector-Quantization</td>
<td>Pascual-Gaspar et al., 2011</td>
</tr>
<tr>
<td>Discrete Cosine Transform</td>
<td>Liu et al., 2014</td>
</tr>
<tr>
<td>Deep learning:</td>
<td></td>
</tr>
<tr>
<td>Neural Networks (MLP)</td>
<td>Malik et al., 2015</td>
</tr>
<tr>
<td></td>
<td>Al-Shoshan, 2006</td>
</tr>
<tr>
<td></td>
<td>Hefny and Moustafa, 2019</td>
</tr>
<tr>
<td>Convolutional Neural Networks (CNNs)</td>
<td>Wu et al., 2019b</td>
</tr>
<tr>
<td></td>
<td>Lai et al., 2020</td>
</tr>
<tr>
<td></td>
<td>Vorugunti et al., 2019</td>
</tr>
<tr>
<td>Recurrent Neural Networks (with LNPS)</td>
<td>Lai and Jin, 2018</td>
</tr>
<tr>
<td>Recurrent Neural Networks (with LSTM)</td>
<td>Tolosana et al., 2018</td>
</tr>
<tr>
<td>Bidirectional Recurrent Neural Network</td>
<td>Nathwani, 2020</td>
</tr>
<tr>
<td>Time-Aligned Recurrent Neural Networks (TA-RNN)</td>
<td>Tolosana et al., 2021a</td>
</tr>
<tr>
<td>Artificial Neural Networks</td>
<td>Nilchiyan et al., 2015</td>
</tr>
<tr>
<td>SVM</td>
<td>Kholmatov and Yanikoglu, 2005</td>
</tr>
<tr>
<td>Gaussian Mixture Model (GMM)</td>
<td>Manjunatha et al., 2016</td>
</tr>
<tr>
<td></td>
<td>Radhika and Gopika, 2015</td>
</tr>
<tr>
<td>Hidden Markov Model (HMM)</td>
<td>Tolosana et al., 2015c</td>
</tr>
<tr>
<td>K-Fold Cross-Validation</td>
<td>Xia et al., 2017</td>
</tr>
<tr>
<td></td>
<td>López-García et al., 2013</td>
</tr>
<tr>
<td></td>
<td>Francis et al., 2015</td>
</tr>
<tr>
<td>Parzen window classifier</td>
<td>Rashidi et al., 2012</td>
</tr>
<tr>
<td>Naïve Bayesian (NB)</td>
<td>Manjunatha et al., 2016</td>
</tr>
<tr>
<td>Nearest neighbor</td>
<td>Manjunatha et al., 2016</td>
</tr>
<tr>
<td>Principal component analysis (PCA)</td>
<td>Manjunatha et al., 2016</td>
</tr>
<tr>
<td>Linear discriminant analysis (LDA)</td>
<td>Manjunatha et al., 2016</td>
</tr>
</tbody>
</table>
2.5 State of the art results comparison

In the previous sections, I discussed the most common and effective methods which were used for each step of online signature verification. Usually, any online signature verification will have a combination of the previously-discussed methods or other methods to build a competitive system. In this section, I compare the final results of the proposed verification systems in the last decade. Table 2.5 shows a comparison between these systems and the verification algorithm used and the database that the systems were applied on. Different approaches and databases were used in these systems. Thus, it was not easy to compare these systems; however, this comparison provides general insight and view for the new researcher. Some previous signature verification competitions have been organized to evaluate the proposed verification methods under similar circumstances and on similar databases, such as SVC2004 [Yeung et al., 2004], BSEC2009 [Houmani et al., 2012], ESRA’2011 [Houmani et al., 2011], SigComp2011 [Liwicki et al., 2011], SigWiComp2013 [Malik et al., 2013], and SigWiComp2015 [Malik et al., 2015]. The new Deep-SignDB database can be a reference database for the researcher to develop and test their verification system on a big database that has signatures with variety of input methods, sampling rates, input devices and more features. This will also provide a fair systems comparisons, more accurate than comparing systems that were applied on different databases. The new signature verification competition 2021 will also provide a benchmark for signature verification systems where signatures can be compared under similar conditions using one approach in reference selections, comparisons, and one database. Thus, I believe, in the near future, there will be fair and informative comparisons for online signature verification.
Table 2.5: A Comparison between methods presented in the last few years, S=stylus, F=finger.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Year</th>
<th>Verification</th>
<th>Database</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Tolosana et al., 2021a]</td>
<td>2021</td>
<td>TA-RNNs</td>
<td>DeepSignDB(S)</td>
<td>3.3%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DeepSignDB(F)</td>
<td>11.3%</td>
</tr>
<tr>
<td>[Chandra et al., 2021]</td>
<td>2021</td>
<td>Machine learning methods</td>
<td>SVC2004</td>
<td>2.62%</td>
</tr>
<tr>
<td>[Nathwani, 2020]</td>
<td>2020</td>
<td>BRNN</td>
<td>SVC2004</td>
<td>8.8%</td>
</tr>
<tr>
<td>[Okawa, 2020]</td>
<td>2020</td>
<td>Mean templates/DTW</td>
<td>SVC2004</td>
<td>1.8%</td>
</tr>
<tr>
<td>[Okawa, 2020]</td>
<td>2020</td>
<td>Mean templates/DTW</td>
<td>MCYT</td>
<td>1.28%</td>
</tr>
<tr>
<td>[Nathwani, 2020]</td>
<td>2020</td>
<td>Single-template strategy</td>
<td>MCYT</td>
<td>8.8%</td>
</tr>
<tr>
<td>[Lai et al., 2020]</td>
<td>2020</td>
<td>CNNs</td>
<td>MCYT</td>
<td>1.7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SVC2004</td>
<td>4.6%</td>
</tr>
<tr>
<td>[Parziale et al., 2019]</td>
<td>2019</td>
<td>SM-DTW</td>
<td>MCYT</td>
<td>3.09%</td>
</tr>
<tr>
<td>[Tolosana et al., 2018]</td>
<td>2018</td>
<td>RNN</td>
<td>BiosecureID</td>
<td>4.75%</td>
</tr>
<tr>
<td>[Xia et al., 2017]</td>
<td>2017</td>
<td>DTW/GMM</td>
<td>MCYT</td>
<td>2.15%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SVC2004</td>
<td>2.63%</td>
</tr>
<tr>
<td>[Lai and Jin, 2018]</td>
<td>2017</td>
<td>RNN/LNPS</td>
<td>MCYT</td>
<td>2.37%</td>
</tr>
<tr>
<td>[Guru et al., 2017]</td>
<td>2017</td>
<td>Threshold</td>
<td>MCYT</td>
<td>4.60%</td>
</tr>
<tr>
<td>[Manjunatha et al., 2016]</td>
<td>2016</td>
<td>Different methods</td>
<td>MCYT</td>
<td>1.10%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SUSIG</td>
<td>1.92%</td>
</tr>
<tr>
<td>[Sharma and Sundaram, 2016c]</td>
<td>2016</td>
<td></td>
<td>MCYT</td>
<td>1.55%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SVC2004</td>
<td>2.73%</td>
</tr>
<tr>
<td>[Fayyaz et al., 2015b]</td>
<td>2015</td>
<td>K-Fold Cross-Validation</td>
<td>SVC2004</td>
<td>0.83%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SUSIG</td>
<td>0.77%</td>
</tr>
<tr>
<td>[Pirlo et al., 2015]</td>
<td>2015</td>
<td>DTW/Threshold</td>
<td>SUSIG</td>
<td>2.125%</td>
</tr>
<tr>
<td>[Francis et al., 2015]</td>
<td>2015</td>
<td>DTW/GMM</td>
<td>MCYT</td>
<td>2.74%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SUSIG</td>
<td>3.09%</td>
</tr>
<tr>
<td>[Fischer et al., 2015]</td>
<td>2015</td>
<td>DTW/Threshold</td>
<td>MCYT</td>
<td>2.74%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SUSIG</td>
<td>3.09%</td>
</tr>
<tr>
<td>[Liu et al., 2014]</td>
<td>2015</td>
<td>Discrete cosine transforms DCT</td>
<td>SVC2004</td>
<td>5.61%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SUSIG</td>
<td>2.98%</td>
</tr>
<tr>
<td>[Nilchiyan et al., 2015]</td>
<td>2015</td>
<td>ANN</td>
<td>SVC2004</td>
<td>2.75%</td>
</tr>
<tr>
<td>[Tolosana et al., 2015b]</td>
<td>2015</td>
<td>DTW</td>
<td>MCYT</td>
<td>4.10%</td>
</tr>
<tr>
<td>[Tolosana et al., 2015c]</td>
<td>2015</td>
<td>HMM/GMM</td>
<td></td>
<td>1.40%</td>
</tr>
</tbody>
</table>
During my work, I reviewed the latest online signature verification results and summarized the most commonly used algorithms in each step of the verification process [Saleem and Kovari, 2020e]. In this chapter, the previously collected algorithms and their combinations are deeply investigated and analyzed to present a systematic evaluation of their effect on the verification accuracy.

I conducted an extensive evaluation of some of the most popular design choices. The resulting 42,336 verifier configurations were tested on five different publicly available databases. Although results for individual methods tested on individual databases are available, these are usually not considered representative results. In addition, the corresponding implementation and additional data are rarely made available. As I aim to aid researchers in the field directly, I have made all the related source code and the research data from over 211,680 experiments available publicly [Kovari et al., 2020].

The chapter is organized as follows: briefly summarizes the previous work in the field. I then introduce and explain the methodological background for the experiment by formally defining the examined preprocessing approaches. After that, I explain the experimental setup in detail and introduce the five databases used to
evaluate the algorithms. Then the results of these experiments are presented and evaluated. The thesis is concluded by summarizing the most important insights that can be gleaned from the results. In addition, the main contributions of the evaluation were used to build a competitive online signature verification system.

3.1 Methodology

3.1.1 Preprocessing

The acquisition device may introduce distortions in the form of noise or variations in sampling frequency, which can significantly influence the accuracy of a verification system. Therefore, my primary goal was to analyze several different preprocessing approaches to eliminate distortions from the captured data. In this thesis, I grouped such works according to the basic geometric transformations that they are based on. The following subsections present these preprocessing approaches in detail.

Let us define an online signature $S$ as a set of $m$ feature vectors $F_i$

$$S = \{F_1, \ldots, F_m\}. \quad (3.1)$$

Each feature vector consists of $n$ data points

$$F = \{f_1, \ldots, f_n\}. \quad (3.2)$$

I will use $X = \{x_1, \ldots, x_n\}$ to reference the feature vector of horizontal coordinates, $Y = \{y_1, \ldots, y_n\}$ for the vertical coordinates, $P = \{p_1, \ldots, p_n\}$ for the pressure values, and $T = \{t_1, \ldots, t_n\}$ for timestamps.

3.1.2 Dynamic Time Warping (DTW)

Many comparison approaches have been used in signature verification systems. In this work, a DTW-based [Fischer et al., 2015] solution is used as it is popular and has shown promising results [Malik et al., 2015]. DTW finds the best non-
linear alignment of two vectors such that the overall distance between them is minimized. In signature verification, the elements of these vectors are the vectors themselves (e.g., \( \{ x_1, y_1, p_1 \ldots \} \)), which means that the DTW can be calculated in two different ways [Mueen and Keogh, 2016]: by defining it as the sum of DTW scores of the dimensions (\( DTW_I \)) or by calculating the distances between the (sub) vector pairs (\( DTW_D \)). As the dimensions in signature verification are strongly coupled, the latter is recommended [Mueen and Keogh, 2016], and I use it in the experiments.

DTW can utilize different distance functions to calculate the similarity value mentioned above. The choice of distance function may affect the accuracy of the verification. This study evaluates two different distance functions: the Euclidean [Danielsson, 1980] and Manhattan distance functions [Menger, 1979].

### 3.1.3 Translation

The DTW algorithm is sensitive to spatial distance [Mueen and Keogh, 2016]. The fact that the location of the signatures differs in each of the samples could negatively impact the accuracy of the verification. Translation can solve this problem by standardizing the location of the signatures, which makes it a commonly used preprocessing step in signature verification [Xia et al., 2017] [Kholmatov, 2003] [Ahrabian and BabaAli, 2019]. It is applied by subtracting a specific value from the data points of the signature \((X, Y)\), which aligns the signature to the exact position needed [Ibrahim et al., 2010].

The translation preprocessing step consists of two parts. First, the new origin should be defined, and subsequently, the translation itself should be executed for all affected features. Let us define the preprocessing step as

\[
f'_i = f_i - o_f,
\]

where \( f'_i \) is the translated feature value of the \( i^{th} \) point of the signature, \( f_i \) is the original feature value of this point, and \( o_f \) is the value of the new origin in the original coordinate system for the given feature.
Chapter 3. Systematic evaluation of preprocessing approaches in online signature verification

There are several alternatives for defining the new origins as well. This study deals with two main approaches that were implemented and applied during the experiments.

The first approach uses the center of gravity of the features as the new origin. The new origin is calculated as

\[ o_f = \frac{\sum_i f_i}{n}. \]  

The second approach uses the minimum values of the features. In two dimensions, the new origin will be the bottom left corner of the signature-enclosing rectangle. The new origin is calculated as

\[ o_f = \min_i f_i. \]  

Although this step can be applied to different features independently, it was applied to all of them in the experiments. When applying it to pressure ($P$), I always assumed that the minimum value is 0, even if there are no actual points with this value. At the end of the translation process, the signatures are located closer to each other than they were initially. The expectation is that this preprocessing helps DTW focus on the shape differences of the different features.

3.1.4 Scaling

Translation was combined with rescaling of the signatures using several normalization techniques. To isolate the beneficial effects of the two preprocessing algorithms, I defined scaling as a transformation independent of translation. Signature scaling has been applied in several previous studies as a preprocessing method (e.g., [Ahrabian and BabaAli, 2019] [Nilchiyan et al., 2015] [Jindal et al., 2018]). It can be applied horizontally, vertically, or in both directions [López-García et al., 2013] [Mohammadi and Faez, 2012]. Researchers even scaled one of the coordinates to [0,1] and scaled the other coordinates such that the ratio between the coordinates did not change [Khan et al., 2006].

One crucial effect of scaling (in conjunction with DTW) is that it can equalize magnitude differences (e.g., between the range of coordinates and the range of
pressure measurements), thereby ensuring that all features contribute to the DTW
distance with a similar weight.

I introduce two approaches to scaling. The first scales the values into a fixed
interval, while the other approach scales the values based on their standard devia-
tion. In both cases, scaling is executed on the spot such that the minimum value
of the scaled features is always fixed.

Let us define "min-max scaling" of a feature $F$ into the interval $[f_{min}, f_{min} + 1]$ as:

$$f_{min} = \min_{i} f_i,$$  \hspace{1cm} (3.6)

$$f_{max} = \max_{i} (f_i - f_{min}),$$  \hspace{1cm} (3.7)

$$f_i' = f_{min} + \frac{f_i - f_{min}}{f_{max}}.$$  \hspace{1cm} (3.8)

Let us define the standard deviation-based scaling of a feature as:

$$f_i' = f_{min} + \frac{f_i - f_{min}}{\text{stdev}(f)},$$  \hspace{1cm} (3.9)

where $f_i'$ is the newly scaled value of $f_i$. In the literature, the two most com-
monly used preprocessing approaches that combine scaling and translation are
$z$-normalization [Auckenthaler et al., 2000] and scaling into the $[0,1]$ range (min–
max normalization). Both of these approaches are included in this study through
combination of the previously introduced translation and scaling methods. In ad-
dition, the distinction between the two allows me to introduce a new combination,
where values are first scaled into the $[f_{min}, f_{min} + 1]$ range and then aligned to
their center of gravity.

When rescaling the pressure, we assumed that the minimum value for the fea-
ture is 0.
Rotation normalization

The DTW algorithm is also sensitive to the orientation of the signatures. Even though signatures are captured within a defined frame, the signature orientation can differ as the paper or tablet may be rotated between signing attempts. Therefore, two similar signatures can be described by different values. Several research reports claim performance benefits from rotation normalization [Rashidi et al., 2012] [Khan et al., 2006] [Wirotius et al., 2004] and [Jindal et al., 2018]. Fayyaz et al. [Fayyaz et al., 2015a] applied it based on orthogonal regression, and Mohammadi et al. [Mohammadi and Faez, 2012] removed the rotation angle using DTW. While the approaches may differ, the basic aim is the same: identifying the main direction of the signature and rotating it until it reaches a horizontal position.

The approach taken here is based on that of Xia et al. [Xia et al., 2017]. This method uses the X and Y feature vectors of a signature. The original values $x_i$ and $y_i$ rotated by $\alpha$ are $x'_i$ and $y'_i$, which can be calculated as follows:

$$x'_i = (x_i \times \cos \alpha) - (y_i \times \sin \alpha),$$  
$$y'_i = (x_i \times \sin \alpha) + (y_i \times \cos \alpha).$$

The rotation angle is calculated as

$$\alpha = \frac{1}{2} \arctan \left( \frac{2I_{xy_{\text{centroid}}}}{I_{y_{\text{centroid}}} - I_{x_{\text{centroid}}}} \right),$$

where $I_{xy_{\text{centroid}}}$, $I_{x_{\text{centroid}}}$, and $I_{y_{\text{centroid}}}$ are the moments of inertia referred to as the reference centroids. See [Xia et al., 2017] for details.

Pen-up strokes

Pen-up strokes describe the movement of the tip of the pen between two signature strokes. Pen-up durations [Robertson and Guest, 2015] [Yanikoglu and Kholmatov, 2009] refer to the time when the pen does not touch the input device. Usually, pen-up strokes take up less time in the orig-
inal signatures, which may help detect the forged signatures. Yanikoglu et al. [Yanikoglu and Kholmatov, 2009] filled in the missing pen-up strokes using synthetic data (e.g., by inserting imaginary points every 30ms on a 100 Hz input device), but different acquisition approaches may handle the pen-up durations in different ways.

The first and most straightforward approach is to remove the pen-up durations. This approach is a trivial transformation, as the points of pen-up strokes always have a pressure value of 0 assigned to them. Such preprocessing may allow us in the benchmark to handle all databases (ones that include pen-up strokes and ones that do not) similarly.

The second approach for unifying the signature data is to fill in the missing data with interpolated points. This study deals with the two most common approaches, linear [Joseph and Wang, 1959] and cubic interpolation [Fritsch and Carlson, 1980]. To do this, I first calculate the average length of the sampling interval using all available signature points.

The signature time feature \((T)\) includes the timestamps of each point. Each pair of consecutive timestamps defines a timeslot, where the length of the timeslot can be calculated as the difference between the two timestamps.

A time slot is (part of) a pen-up stroke when its length is longer than the median of the timeslot length. The detected pen-up intervals were filled with new interpolated points, with a pairwise distance that equals the median timeslot length. The values for other features at the new timestamps were calculated either by linear or by cubic spline interpolation.

### 3.1.7 Resampling

Resampling the signature may also be a viable preprocessing step. López-García et al. [López-García et al., 2013] transformed the average signals to equally spaced 256-points using linear interpolation. Resampling the number of points to meet the reference signature point count was also a strategy used by Nilchiyan et al. [Nilchiyan et al., 2015]; they used cubic spline interpolation for the purpose, similar to [Ibrahim et al., 2010].
Chapter 3. Systematic evaluation of preprocessing approaches in online signature verification

The main goal of these methods is to standardize the sample count \( (n) \) of a signature. My approach calculates a consistent time slot length for the signature based on the expected number of samples and the original length of the signature and estimates the new sample values based on the original data using an interpolation technique.

Let \( T = \{t_1, t_2, t_3, \ldots, t_n\} \) represent the timestamps corresponding to the original sample points, and \( n^* \) be the expected number of samples. The new time slot length can be calculated as

\[
t = \frac{t_n - t_1}{n^* - 1}.
\]

(3.13)

The original timestamps may be replaced with new ones with respect to \( t'_i = t'_{i-1} + t, \ t'_1 = t_1 \). The values of the other features are calculated at the new timestamps by interpolation. Two interpolation types were used in our experiments: cubic and linear.

3.2 Experimental setup

A signature verifier was created for all linear combinations of the previously introduced methods, including some additional choices such as feature selection or the distance function used in conjunction with DTW. These configurations were tested on five public databases. Figure 3.1 summarizes all the available choices in the system.

The resulting 211680 combinations were executed on two computer clusters of the Budapest University of Technology and Economics, using single-core virtual instances. The total calculation time exceeded 30000 core-hours. The resulting data were stored in MongoDB, reaching a size of 86 GB. In addition to the classification results and execution information, all the DTW distances were calculated for all affected signature pairs, making the further evaluation of these methods much simpler and cheaper. The dataset and source code of the benchmark framework are freely available [Kovari et al., 2020].
This section summarizes all the configuration choices not directly related to the preprocessing steps.

3.2.1 Datasets

Five popular public datasets were selected for the experiment. The datasets of the SVC2004, MCYT100, the Dutch and Chinese datasets from the Signature Verification Competition for Online and Offline Skilled Forgeries 2011 (SigComp’11 Dutch, SigComp’11 Chinese), and the SigWiComp2015. All of these datasets contain skilled forgeries. Table 2.1 summarizes the relevant properties of these datasets.

3.2.2 Feature selection

The above datasets contained some of the following information about individual signatures:

- X: the horizontal positions of sample points.
- Y: the vertical positions of sample points.
- T: the timestamps when the sample points were captured.
- P: the pressure values of the pen on the input device.
- Azimuth (Az): clockwise rotation of pen around the z-axis.
• Altitude (Alt): angle of pen-upward toward the positive z-axis.

Although all datasets contain spatial information and pressure, only two of them include altitude and azimuth. In the interest of consistent results and conclusions, I decided to create only configurations limited to X, Y, P, and T when available. Some studies suggest that these features may not be of equal importance [Dolfing et al., 1998]; all possible combinations, namely \( \{X\}, \{Y\}, \{P\}, \{X, Y\}, \{X, P\}, \{Y, P\}, \) and \( \{X, Y, P\} \) have been evaluated.

### 3.2.3 Test set selection (split)

For each experiment, the datasets were divided into two disjoint sets (training and testing sets). The training set contained ten genuine signatures (reference signatures) for each signee that were used to evaluate the test set samples (questioned signatures). The test set contained all the remaining signatures for each signee (both genuine and forged). A popular approach for selecting the training set is to take each signee’s first few genuine signatures from the database. Although this approach makes it easier to reproduce the experiment, it is also highly biased.

To show the degree of bias, I introduced three additional new deterministic approaches for training set selection. The following four approaches have been used in our experiments to select the test set from an ordered list of genuine signatures of a given signee:

- **first 10**: use the first ten signatures.
- **last 10**: use the last ten signatures.
- **odd 10**: use the first ten signatures with odd indexes.
- **even 10**: use the first ten signatures with even indexes.

### 3.2.4 Classification

After having the reference signatures selected, the preprocessing steps executed, I calculated the DTW distances for all reference-questioned signature pairs. The classifier used to make decisions is a variant of the one-class nearest neighborhood classifier [Khan, 2010]. I compared the average DTW distances between a
questioned signature and all references to a signee-specific threshold to decide the origin of a signature. Signatures above the threshold are classified as forgeries; all other signatures are accepted as genuine. In the experiments, the threshold was set to achieve similar false acceptance and false rejection rates, thereby approximating the equal error rate (EER), which is used to measure the accuracy of our configuration.

### 3.3 Experimental results

There are a total of 10 configuration parameters that affect the results in different ways. Moreover, there may be situations where a transformation, such as rotation normalization, can improve the verifiers’ accuracy, while it may worsen it in other cases. In the following subsections, I systematically eliminate the configurations that yield consistently worse results compared to their counterparts by going through these parameters.

#### 3.3.1 Effects of training set selection

The results show that the training set selection algorithm impacts the verification results. Figure 3.2 summarizes the average EER for each of the datasets.

![Figure 3.2: Effect of training set selection strategy on the average EER.](image)
The execution of the same configurations using different training set results in relevant differences to the attainable EER values. The most significant difference (8.85%) can be seen in the database SVC2004. Here, the best average ERR (18.07%) was achieved using the first signatures with odd indices in the training set, while the worst results (26.92%) were achieved by using the first ten signatures for training.

The intuitive reason behind these differences is that the first and last ten reference signatures of the SVC2004 database were captured separately. In SVC2004, each data contributor was asked to contribute genuine signatures in two separate sessions, ten signatures in the first session and ten signatures in the second session, which was at least one week after the first session [Yeung et al., 2004]. A mixture of these signatures allows for a more exact and robust classification. Unfortunately, the available data on the other databases do not allow me to generalize the conjecture.

In the following, I always take the average of the EER rates reached by these four approaches to obtain more general results. In the subsequent discussion, every four configurations that differ only in the reference selection method are replaced with one that has no selection aspect. The corresponding EER of these new records was calculated as the average of the original four EERs.

### 3.3.2 Effects of resampling

In addition to the original sampling rate, four different sample counts (50, 100, 500, and 1000) were used to achieve both upsampling and downsampling. To evaluate the results, I compared the results of individual signature verifier configurations that differed only in their sample counts. In the experiments, resampling the signatures to a fixed number of points always increased the error rate in four out of five databases. In the case of the fifth database (MCYT100), the error rate also increased in 99% of all the cases, compared to the verification results of the original sampling rates. The results were similar for both linear and cubic resampling.

Based on the results, resampling to a fixed number of samples in this manner does not seem to be a viable option. To simplify the evaluation of the results, I
Table 3.1: Percentage of all Configurations, Where Removing the Pen-up Strokes Resulted in a Lower EER

<table>
<thead>
<tr>
<th>Database</th>
<th>Filtering improved EER (all configs)</th>
<th>Filtering improved EER (top 5%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCYT-100</td>
<td>13.27%</td>
<td>77.78%</td>
</tr>
<tr>
<td>SigComp’11 (Dutch)</td>
<td>29.59%</td>
<td>100.00%</td>
</tr>
<tr>
<td>SigComp’11 (Chinese)</td>
<td>38.78%</td>
<td>100.00%</td>
</tr>
</tbody>
</table>

removed these records from the result set. The following results were obtained for the configurations without resampling.

### 3.3.3 Handling of pen-up strokes

The five databases handle pen-up strokes differently. SVC2004 and the SigWiComp2015 datasets do not contain pen-up strokes, while MCYT100 and SigComp’11 Dutch and Chinese do include them in the signature descriptions.

First, all pen-up points were removed from the latter three using the previously discussed filtering method to allow a uniform handling of the databases. To assess the effects of this operation, I compared pairwise verification results of the non-filtered and filtered configurations. While the results may seem inconclusive at first, if I limit my evaluation to the best 5% of configurations (for each of the three affected databases), then it can be seen that removing pen-ups is always beneficial for the SigComp’11 Dutch and Chinese datasets, and it also has a positive effect in 78% of all the cases for the MCYT100 dataset. Table 3.1 summarizes these results.

The idea behind only considering the best 5% is that there are configurations that reached unacceptably high EER because of the negative effects of some other preprocessing methods. The results of these weaker configurations can distort tendencies, as the filtering method may have other effects on these than on configurations with other (better-chosen) preprocessing approaches.

After the previous filtering operation, all five databases are clear of pen-up points, which can then be added synthetically with the method described in Section 3.1.6. Table 3.2 shows the percentage of all configurations, where the new pen-up points improved the final EER compared to their configuration pair where pen-up
Chapter 3. Systematic evaluation of preprocessing approaches in online signature verification

Table 3.2: Percentage of all Configurations, Where Inserting Synthetic Pen-up Strokes Resulted in a Lower EER

<table>
<thead>
<tr>
<th>Database</th>
<th>FillGap linear</th>
<th>FillGap cubic</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC2004</td>
<td>0%</td>
<td>6.7%</td>
</tr>
<tr>
<td>MCYT-100</td>
<td>12.5%</td>
<td>12.5%</td>
</tr>
<tr>
<td>SigComp'11 (Dutch)</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>SigComp'11 (Chinese)</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>SigWiComp2015</td>
<td>18.8%</td>
<td>18.8%</td>
</tr>
</tbody>
</table>

points were not used. The results show that these simple transformations had a negative effect in a majority of cases.

In conclusion: I compared three approaches for handling pen-up strokes, namely using the original pen-up strokes, using only pen-down strokes, and interpolation-generated pen-up strokes. Based on the discussed results, I suggest using only pen-down strokes. Accordingly, only the results without any pen-up strokes are considered further.

It is worth mentioning that this could result differently if the signatures are acquired using a different approach. For example, in-air signatures [Fang et al., 2017] are acquired using video records, which in this case capture the pen-ups differently. In the used databases, pen-ups are captured as one point with zero value of the pen-down feature. Therefore, the hesitation of the signer will not be captured. Thus, in this situation, no significant effect can be extracted from the pen-ups in this regard.

3.3.4 Distance function within DTW

Previous studies have already evaluated the Manhattan distance as an alternative distance function to the most commonly used Euclidean distance. Some of them suggested the use of the former [He et al., 2019] [Sae-Bae and Memon, 2013], while others recommend the latter [Arora et al., 2015]. My results show that the Manhattan distance may be better by a small margin.

Although, in many cases, both distance functions reached similar EER values, Manhattan was more efficient in cases where the results differed. In Table 3.3, the
effectiveness of the distance functions is compared in terms of two aspects. The first is the percentage of cases in which a distance function improves the equal error rate compared to the other. The second aspect is the quantity of this improvement. The results show that the ratio of cases when the usage of Manhattan distance improved the equal error rate compared to the usage of Euclidean distance was higher than its inverse. In addition, the use of Manhattan distance resulted in more significant improvements.

In conclusion: I recommend using Manhattan distance over Euclidean distance, and I continue filtering the discussed configurations based on this distance function. Therefore, I disregard the configurations using Euclidean distance for the remainder of this chapter.

### 3.3.5 Effects of translation and scaling

Although the three translation options (none, to COG, to zero) and three scaling options (none, \([\text{min}, \text{min}+1]\), stdev) produce nine combinations, not all of them are meaningful. As pressure uses a scale different from \(X\) and \(Y\) features, it is inconsequential to compare them without rescaling them first. Therefore, two of the nine pairs that do not use scaling but use translation were removed at the beginning of the experimental evaluation. The error rates of the remaining combination pairs are summarized in Table 3.4. The best approaches take advantage of both types of translation and scaling algorithms. Two combinations in fact correspond to popular approaches, as the translation to the COG and normalization by standard deviation is \(z\)-normalization (or \(z\)-score), and translation to zero and scaling.
Table 3.4: Average EER of Configurations Using the Given Translation and Scaling Methods for Preprocessing

<table>
<thead>
<tr>
<th>Preprocessing</th>
<th>Translation</th>
<th>Scaling</th>
<th>Database</th>
<th>svc2004</th>
<th>mcc100</th>
<th>SigComp’11 (Dutch)</th>
<th>SigComp’11 (Chinese)</th>
<th>SigWiComp2015</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>19.06%</td>
<td>31.65%</td>
<td>3.79%</td>
<td>14.65%</td>
<td>44.00%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24.79%</td>
<td>31.20%</td>
<td>3.51%</td>
<td>16.02%</td>
<td>42.91%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11.61%</td>
<td>5.97%</td>
<td>3.85%</td>
<td>3.85%</td>
<td>11.33%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8.54%</td>
<td>3.75%</td>
<td>2.38%</td>
<td>2.80%</td>
<td>10.59%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>26.03%</td>
<td>32.07%</td>
<td>3.71%</td>
<td>16.48%</td>
<td>10.59%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16.36%</td>
<td>9.45%</td>
<td>4.65%</td>
<td>5.98%</td>
<td>25.14%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8.79%</td>
<td>3.49%</td>
<td>2.34%</td>
<td>2.30%</td>
<td>9.62%</td>
</tr>
</tbody>
</table>

To [min,min+1] is actually min-max feature scaling. Both approaches are popular within the field. In addition to these two combinations, a hybrid approach seems feasible. When values are scaled to the [min,min+1] range and then translated to the COG, the verification usually yields low error rates. We call this the "centered min-max" approach.

If we check all the remaining verifier configurations that use one of the previous three approaches, we can compare them pairwise for each of the configurations. The results (summarized in Table 3.5) show that the commonly used min-max approach is bested by both z-normalization and the centered min-max normalization. The remaining six approaches almost always underperformed these three; therefore, I only kept the configurations with the three selected translation-scaling combinations for the rest of the evaluation.

### 3.3.6 The effects of rotation normalization

The rotation normalization of the signatures did not improve the accuracy of the verification for a majority of the remaining configurations. Moreover, the use of rotation normalization resulted in higher EER values than the original angle of signatures in a majority of the cases. Figure 3.3 shows the effects of rotation nor-
Table 3.5: Comparison of Three Normalization Approaches for the Remaining Configuration Triplets

<table>
<thead>
<tr>
<th>Database</th>
<th>centered min-max better than min-max</th>
<th>z-normalization better than min-max</th>
<th>z-normalization better than centered min-max</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC2004</td>
<td>100.00%</td>
<td>100.00%</td>
<td>42.86%</td>
</tr>
<tr>
<td>MCYT-100</td>
<td>100.00%</td>
<td>100.00%</td>
<td>71.43%</td>
</tr>
<tr>
<td>SigComp’11 (Dutch)</td>
<td>100.00%</td>
<td>100.00%</td>
<td>35.71%</td>
</tr>
<tr>
<td>SigComp’11 (Chinese)</td>
<td>100.00%</td>
<td>100.00%</td>
<td>85.71%</td>
</tr>
<tr>
<td>SigWiComp2015</td>
<td>64.29%</td>
<td>78.57%</td>
<td>71.43%</td>
</tr>
</tbody>
</table>

malization compared to the non-normalized configurations. The columns show the ratios of cases when the usage of rotation normalization improved the verification accuracy, when it had no effect and when it resulted in worse verification results than the same configuration without rotation. Note that this preprocessing technique did not improve the results in three out of the five databases in any of the cases.

Based on the experimental results, the rotation angle of signatures seems to be more a trait than distortion, and normalizing it increases the error rates. Therefore, I remove the configurations using rotation normalization and study only the remaining configurations.

Figure 3.3: Effects of rotation normalization on the average EER.
As part of the preprocessing benchmark, the implementation and evaluation of the rotation normalization algorithms and the training set selection were conducted by my fellow researcher, Cintia Lia Szucs. The results are also introduced in this work as they form an essential part of the whole study.

### 3.3.7 Feature selection

At this point, all of the configurations are limited to a carefully selected set of fixed preprocessing methods. The only remaining variable is the set of features to consider for verification. Figure 3.4 shows the average EER values for different feature sets grouped by the database. It can be seen that the error rates are different across the databases, as well as the usability of different feature sets. For example, the usage of pressure yields acceptable results in the SigComp’11 Chinese and Dutch datasets, but it is less efficient with the SigWiComp2015 dataset. Because of the differences between the databases, one cannot predict which individual feature has the best discriminative power; however, several single features can be grouped. The results confirm that DTW is useful in utilizing multiple features; therefore, the usage of all three features ($X$, $Y$, $P$) together yields the highest accuracy compared to other feature sets in all the discussed databases. Therefore, I recommend using $X$, $Y$, and $P$ together to obtain the best results. It should
be noted that although the average EER of the remaining configurations may be lower for other feature combinations, the best EER was always achieved when the $XYP$ feature combination was used.

### 3.3.8 Proposed system(s):

According to previous observations, I can define several configurations that yield near-optimal results for most databases. Such a verifier should always consider all three features ($X$, $Y$, and $P$), use the Manhattan distance to calculate the $DTW_D$ score of signature pairs, and should remove the pen-up durations without filling these with artificial data. Resampling and rotation normalization should also be omitted. The best location and scale normalization techniques were $z$-normalization and centered min-max normalization. As non-centered min-max normalization is also widespread in the field, I show the verification results for all three approaches compared with recently published results in Table 3.6 for the SVC2004 dataset and Table 3.7 for the MCYT100 dataset. In Table 3.8, I compare the results with the recently published results that were derived from datasets of the SigComp’11 and SigWiComp2015 competitions.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Year</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Gruber et al., 2009]</td>
<td>2009</td>
<td>6.84%</td>
</tr>
<tr>
<td>[Wang et al., 2011]</td>
<td>2011</td>
<td>6.65%</td>
</tr>
<tr>
<td>[Barkoula et al., 2013]</td>
<td>2013</td>
<td>5.33%</td>
</tr>
<tr>
<td>min-max</td>
<td></td>
<td>4.45%</td>
</tr>
<tr>
<td>[Rashidi et al., 2012]</td>
<td>2012</td>
<td>3.61%</td>
</tr>
<tr>
<td>[Yeung et al., 2004]</td>
<td>2004</td>
<td>2.84%</td>
</tr>
<tr>
<td>$z$-normalization</td>
<td></td>
<td>2.83%</td>
</tr>
<tr>
<td>[Chandra et al., 2021]</td>
<td>2021</td>
<td>2.62%</td>
</tr>
<tr>
<td>[Hu et al., 2019]</td>
<td>2019</td>
<td>2.5%</td>
</tr>
<tr>
<td>[Jia et al., 2019]</td>
<td>2019</td>
<td>2.39%</td>
</tr>
<tr>
<td>[Lai et al., 2017]</td>
<td>2017</td>
<td>2.37%</td>
</tr>
<tr>
<td>centered min-max</td>
<td></td>
<td>2.33%</td>
</tr>
</tbody>
</table>
Table 3.7: Comparison of Results on MCYT100

<table>
<thead>
<tr>
<th>Reference</th>
<th>Year</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Kholmatov and Yanikoglu, 2009]</td>
<td>2009</td>
<td>7.8%</td>
</tr>
<tr>
<td>[Nanni and Lumini, 2008]</td>
<td>2008</td>
<td>5.2%</td>
</tr>
<tr>
<td>[Sae-Bae and Memon, 2014]</td>
<td>2014</td>
<td>4.02%</td>
</tr>
<tr>
<td>[Rua and Castro, 2012]</td>
<td>2012</td>
<td>2.85%</td>
</tr>
<tr>
<td>[Tang et al., 2017]</td>
<td>2020</td>
<td>2.25%</td>
</tr>
<tr>
<td><strong>min-max</strong></td>
<td></td>
<td><strong>1.92%</strong></td>
</tr>
<tr>
<td>[Sharma and Sundaram, 2016a]</td>
<td>2016</td>
<td>1.55%</td>
</tr>
<tr>
<td><strong>z-normalization</strong></td>
<td></td>
<td><strong>1.35%</strong></td>
</tr>
<tr>
<td>centered min-max</td>
<td></td>
<td>1.35%</td>
</tr>
<tr>
<td>[Ibrahim et al., 2010]</td>
<td>2010</td>
<td>1.09%</td>
</tr>
</tbody>
</table>

3.4 Chapter summary

I have described, conducted, and evaluated an experimental evaluation of assessing several design choices for an online ASV system. Unlike those of most existing studies, the results are not limited to positive findings but include all the negative results as well, and the source code of the algorithms can be used to reproduce and evaluate each of our findings. I iteratively evaluated and excluded preprocessing approaches and narrowed down the set of the most efficient approaches to a handful of configurations (Figure 3.5). The most important observations were the following:
Table 3.8: Comparison of Results on the SigComp’11 and SigWiComp2015 Databases

<table>
<thead>
<tr>
<th>Database</th>
<th>Reference/(ID)</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>SigComp11(Dutch)</td>
<td>[Liwicki et al., 2011]/(7)</td>
<td>7.07%</td>
</tr>
<tr>
<td>SigComp11(Dutch)</td>
<td>[Liwicki et al., 2011]/(1)</td>
<td>6.51%</td>
</tr>
<tr>
<td>SigComp11(Dutch)</td>
<td>[Liwicki et al., 2011]/(4)</td>
<td>3.73%</td>
</tr>
<tr>
<td>SigComp11(Dutch)</td>
<td>[Liwicki et al., 2011]/(5)</td>
<td>3.65%</td>
</tr>
<tr>
<td>SigComp11(Dutch)</td>
<td>[Riesen and Schmidt, 2019]</td>
<td>3.24%</td>
</tr>
<tr>
<td>SigComp’11 (Dutch)</td>
<td>min-max</td>
<td>1.58%</td>
</tr>
<tr>
<td>SigComp’11 (Dutch)</td>
<td>z-normalization</td>
<td>0.83%</td>
</tr>
<tr>
<td>SigComp’11 (Dutch)</td>
<td>centered min-max</td>
<td>0.72%</td>
</tr>
<tr>
<td>SigComp11(Chinese)</td>
<td>[Liwicki et al., 2011]/(6)</td>
<td>17.6%</td>
</tr>
<tr>
<td>SigComp11(Chinese)</td>
<td>[Liwicki et al., 2011]/(1)</td>
<td>15.19%</td>
</tr>
<tr>
<td>SigComp11(Chinese)</td>
<td>[Liwicki et al., 2011]/(7)</td>
<td>14.68%</td>
</tr>
<tr>
<td>SigComp11(Chinese)</td>
<td>[Liwicki et al., 2011]/(4)</td>
<td>6.83%</td>
</tr>
<tr>
<td>SigComp’11 (Chinese)</td>
<td>min-max</td>
<td>1.11%</td>
</tr>
<tr>
<td>SigComp’11 (Chinese)</td>
<td>z-normalization</td>
<td>0.86%</td>
</tr>
<tr>
<td>SigComp’11 (Chinese)</td>
<td>centered min-max</td>
<td>0.64%</td>
</tr>
<tr>
<td>SigWiComp2015</td>
<td>z-normalization</td>
<td>5.04%</td>
</tr>
<tr>
<td>SigWiComp2015</td>
<td>min-max</td>
<td>5.02%</td>
</tr>
<tr>
<td>SigWiComp2015</td>
<td>centered min-max</td>
<td>5.01%</td>
</tr>
</tbody>
</table>

While the commonly used approach of using the first ten genuine signatures as a training set is easily reproducible, it is highly biased and should be avoided, especially when the signatures are sorted by their acquisition time.

Resampling signatures to a fixed number of samples with linear or cubic interpolation only improved the result in a negligible number of cases, while in all other cases, it had a negative impact. Therefore, I do not recommend using this signature verification approach when the capturing device and circumstances are similar.

Existing pen-up strokes did not significantly contribute to improving the classification results in the examined databases. Filling in pen-up strokes with synthetic interpolated data mostly affected the results in a negative manner.

The Manhattan and Euclidean distance-based DTW approaches had the same classification results in approximately half of the investigated cases. In the rest of
the cases, Manhattan distance produced more often better results and decreased the error rate by approximately twice as much as the Euclidean distance.

Experimental data showed that $z$-normalization was superior to traditional min-max normalization. I examined a novel combination called centered min-max normalization, which yielded competitive results comparable to that of $z$-normalization.

Rotation normalization worsened the classification results in over 80% of the total cases and could not produce any performance improvement in the three datasets. The data suggest its usage to be more counterproductive in cutting-edge classifiers.

The individual discriminatory powers of the features X, Y, and P differ significantly in the five databases. The results show that $DTW_D$ was efficient in compensating these differences without prior knowledge of the database and achieved the best accuracy by using all three features.

Based on the experimental data, the best signature verifier configuration uses $z$-normalization to compare X, Y, and P without pen-up strokes using $DTW_D$ and does not benefit from any other preprocessing approaches.

I introduced the centered min-max normalization, which may be an alternative to $z$-normalization, and yielded comparable results. These two configurations produced competitive EER values to that of more complex approaches in recent publications.
Sampling frequency based online signature verification system

Online signature verification considers signatures as time sequences of different measurements of the signing instrument. These signals are captured on digital devices and therefore consist of a discrete number of samples. To enrich or simplify this information, several verifiers employ resampling and interpolation as preprocessing steps to improve their results; however, their design decisions may be difficult to be generalized. This chapter investigates the direct effect of the sampling rate of the input signals on the accuracy of online signature verification systems without using interpolation techniques and presents a signer-dependent online signature verification system.

4.1 The effect of sampling rate on the verification accuracy

In online signature verification, a device (like a tablet or a camera) is used to acquire the signature as a function of time for each feature that can be captured. The capturing typically happens at frequencies between 75 Hz and 200 Hz. This thesis aims to study the direct effect of the sampling frequency and the number of sample points on a simple signature verification system. One of the expectations
was that the error rate would be decreased when the sampling frequency increases because more points should provide more information about the signatures. It was also expected that the error rate might reach a minimum level at a sufficiently high frequency.

I conducted thousands of measurements on five different public signature databases, and the results of the experiments showed a different behavior from the mentioned expectations. The relation between the sampling frequency and the error rate was not monotonous in the majority of the cases; however, the error rate had a local minimum. Moreover, this local minimum was achieved in a similar range for several databases. These results will be explained in detail in the subsequent sections.

### 4.1.1 Related work

Sampling is applied when a digital device is used to acquire an analog signal by recording it at a specific frequency. If this sampling frequency is sufficiently large, human perception cannot notice the difference between digitized and analog information. Although the input devices used for signature acquisition have sampling rates as high as 200 Hz, it does not mean that they will provide better verification performance. The first study on signature frequencies [Plamondon and Lorette, 1989a] stated that signature signals have a maximum frequency ($f_s$) of 20 Hz – 30 Hz. Throughout the past three decades, several papers have been published and dealt with the subject. Another study suggested that, as the number of harmonics in handwriting is low, 5 Hz is sufficient to provide the most important frequency components and 10 Hz for all of them, whereas to be able to apply some filters for noisy data, one needs a frequency range of 10 Hz – 37 Hz [Teulings and Maarse, 1984].

The Nyquist rate or frequency [Landau, 1967] is the minimum rate at which a finite bandwidth $B$ signal needs to be sampled to retain all of the information. The sampling frequency should be at least double the highest frequency contained in the signal. The Shannon–Nyquist sampling theorem [Jerri, 1977] guarantees that any signal whose Fourier transform is supported on this bandwidth limit can be entirely reconstructed from the discrete-time signal as long as the frequency rate
is at least double this bandwidth limit, as illustrated by the following equation [Romanov and Ordentlich, 2019]:

\[ f_s > 2B \]  

(4.1)

In our case, a frequency range of 40 Hz – 60 Hz should be sufficient to contain all the signature information without providing redundant information. Although this is a general theory for signal sampling, it helps us understand and analyze the results and explain some behaviors.

Martinez-Diaz et al. studied the effects of sampling rate and interpolation in an HMM-based verification system on a single database [Martinez-Diaz et al., 2007]. The signatures were down-sampled to 25 and 50 Hz and then up-sampled to 100 Hz (the original frequency of the input device) by using Catmull-Rom [Dodgson, 1997] and linear interpolation schemes. Their results showed that the accuracy could be improved by using re-sampling and interpolation together.

The above results show the benefits of using re-sampling and interpolation but do not show the direct effect of the frequency itself, and they cannot be generalized to other databases and verification systems. That is why, in this work, I used different approaches for sampling and preprocessing and many different sampling rates on five different databases.

Vivaracho-Pascual et al. proposed a low-cost approach to online signature recognition based on length normalization. Although their work was about signature recognition, not verification, it showed that, for the MCYT database, it is possible to reduce the number of signature points without performance loss [Vivaracho-Pascual et al., 2009]. In offline signatures, image resolution is similar to sampling frequency for online signatures. Vargas et al. investigated the effect of image resolution on the verification performance; they used images with a resolution of as high as 600 dpi to as low as 45 dpi. Their results showed that a resolution of 150 dpi offers a good trade-off between performance and image resolution [Vargas et al., 2007].

Furthermore, Vatavu studied the effect of sampling rate on the performance of template-based gesture recognizers using down-sampling and a DTW approach.
His results showed that six sampling points are sufficient for Euclidean and angular recognizers to provide high-performance [Vatavu, 2011].

### 4.1.2 Proposed verification system

The previous results were obtained mainly by using some interpolation approach. It cannot be clearly stated whether the changes in the verification or recognition accuracy should be attributed to the resampling or the interpolation itself. In this thesis, interpolation was not used to avoid its effect on the results. In the following section, the verification system used and the experimental protocol are discussed in detail.

I created a simple signature verification system and evaluated it with different preprocessing approaches on several databases to support my conclusions. This will provide a large variation in the experimental work and eliminate some other factors that may affect the system’s accuracy.

Five different databases were used in this work to avoid any data-dependent results: The Signature SVC2004, the MCYT-100, the Dutch and Chinese subsets of the SigComp’11) database, and the SigWiComp’15 database (see Table 2.1).

Preprocessing is important to improve the similarity measurement accuracy. Scaling, translation, and z-normalization methods were chosen for preprocessing purposes. Signature scaling may be used to resize all signatures by multiplying all the points by a certain ratio to keep the signature in a specific range (see Equation 4.2).

\[
\hat{x}(i) = x_{newMin} + \frac{x(i) - x_{oldMin}}{x_{oldMax} - x_{oldMin}} \times (x_{newMax} - x_{newMin})
\]

(4.2)

In the case of translation, all signature points were shifted by a given vector. In this thesis, I used translation to move the center of gravity of the signatures to the origin using the following equation:

\[
\hat{x}(i) = x(i) - \mu_x
\]

(4.3)

Normalization to zero mean and unit of energy (Z-normalization) aims to transform all elements of the input vector into an output vector where its mean approximately 0, and the standard deviation is around 1. Here is the formula used in this
work for the $z$-normalization:

$$\hat{x}(i) = \frac{x(i) - \mu}{\sigma}, \text{ where } i \in N$$ (4.4)

The three preprocessing methods and their combinations were used in the verification systems.

I chose a compilation of the horizontal ($X$) and vertical positions ($Y$) and the pressure ($P$) as they were available in all the databases used. Five different combinations of features were tested: $X$, $Y$, $P$, $XY$, and $XYP$.

After conducting the previous steps, the signatures were ready to be verified. For each signer, ten genuine signatures were chosen to act as references. These signatures were used to calculate a similarity threshold for the verification using DTW, which can be used with different distance measurement algorithms; here, the Euclidean distance was used. According to the Euclidean distance formula, the distance between two points in the plane with the coordinates $(a_1, b_1)$, and $(a_2, b_2)$ is given by

$$\text{dist}((a_1, b_1), (a_2, b_2)) = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}$$ (4.5)

After doing the preprocessing for all the signatures, a threshold for each signer was calculated. When a new signature was being tested, the average distance from the reference signatures was calculated and then compared with the threshold. If it was equal to or lower than the threshold, the signature was classified as genuine; otherwise, it was classified as forged. The EER metric was used for the system evaluation.

As the choice of reference signatures might directly affect the results, I employed two different deterministic strategies for it. In the ”first” strategy, the first ten genuine signatures of the signer were chosen as references. In the ”even” strategy, the first ten genuine signatures with an even index (2,4,6,8,...) were used.
4.1.3 Signature sampling

To test the effect of the sampling rate and the number of sample points on the verification accuracy, the verification steps using different sampling rates in each test were applied. Thus, in each test, some points were skipped to reduce the sampling rate and signature points. The test was initiated with the initial sampling rate, and then, in each iteration, some of the points were skipped. The number of iterations depends on the average number of sample points in the database. For SVC2004, MCYT-100, and SigWiComp’15, 20 iterations were used, whereas for SigComp’11 (Dutch and Chinese), 40 iterations were used. These iterations provided the tests for sampling rates between 5 Hz and 200 Hz. The ranges of the sampling rates and the average number of signature points tested were as the following: SVC2004, 5 Hz – 100 Hz (10–208 points); MCYT-100, 5 Hz – 100 Hz (21–440 points); SigWiComp’15, 3 Hz – 75 Hz (6–125 points); SigComp’11 (Dutch), 5 Hz – 200 Hz (24–978 points); and SigComp’11 (Chinese), 5 Hz – 200 Hz (19–792 points).

4.1.4 Experimental protocol

The applied signature verification system uses different databases, features, preprocessing, and verification methods, which provides several results that can help generalize the results regardless of the effect of the specific methods chosen. With five databases, four preprocessing methods, five combinations of features, and two different criteria for choosing the training set, I was able to test the results using 200 different configurations (see Figure 4.1). The 40 different combinations of the verification system (for each database) were then applied using different sampling rates. As discussed earlier, the number of iterations used was different for the databases. Overall, 5600 different tests were applied in these iterations. For each configuration, the results were evaluated and visualized to study the exact effect in all cases. Algorithm 4.1 pseudo-code describes the experimental protocol of the work.
Figure 4.1: Combinations of the applied verifiers.

Algorithm 4.1 Pseudo-code of the experimental protocol

```
for database in Databases do
    while Combinations != NULL do
        x=1
        i = iteration number // 20 or 40
        s= d.freq // device sample frequency
        for x<=i do
            s=d.freq/x
            resample the signatures using sampling rate s
            calculate EER
            x=x+1
        return EER
```

4.1.5 Experimental results

After applying all the previous combinations at different sampling rates and numbers of signature points, I analyzed and studied the results to determine the effect of using different sampling rates on the accuracy of a verification system. The best 200 results in each combination are shown in Table 4.1. I selected the sampling rate for each configuration where the lowest EER was achieved.

4.1.5.1 EER expected vs. actual behavior

The results showed that we could obtain better results by decreasing the sampling rate and the average sample points of the databases in most of the combinations. The expected behavior was that the accuracy would be increased if the sampling
Table 4.1: Best sampling frequencies, sample counts, and EER for each combination for each database.

<table>
<thead>
<tr>
<th>Combination</th>
<th>MCYT</th>
<th>SigComp’11 (Dutch)</th>
<th>SigComp’11 (Chinese)</th>
<th>SVC2004</th>
<th>SigWiComp’15 (German)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Frequency</td>
<td>Average points</td>
<td>Frequency</td>
<td>Average points</td>
<td>Frequency</td>
</tr>
<tr>
<td>• • • • X</td>
<td>25 110 4.1%</td>
<td>25 122 3.0%</td>
<td>50 198 2.4%</td>
<td>50 104 8.9%</td>
<td>75 125 9.1%</td>
</tr>
<tr>
<td>• • • • Y</td>
<td>50 220 3.4%</td>
<td>40 188 5.0%</td>
<td>25 99 1.8%</td>
<td>100 208 3.5%</td>
<td>75 125 7.8%</td>
</tr>
<tr>
<td>• • • • P</td>
<td>20 88 4.9%</td>
<td>50 244 2.0%</td>
<td>25 99 1.8%</td>
<td>100 208 7.3%</td>
<td>75 125 16.9%</td>
</tr>
<tr>
<td>• • • • XY</td>
<td>100 440 1.8%</td>
<td>25 122 1.2%</td>
<td>25 99 1.0%</td>
<td>33 69 3.1%</td>
<td>75 125 4.1%</td>
</tr>
<tr>
<td>• • • • XY’</td>
<td>30 220 1.7%</td>
<td>28 140 1.6%</td>
<td>18 72 2.6%</td>
<td>33 69 9.3%</td>
<td>75 125 3.5%</td>
</tr>
<tr>
<td>• • • • XY P</td>
<td>50 220 1.7%</td>
<td>28 140 0.3%</td>
<td>200 792 1.8%</td>
<td>100 208 1.4%</td>
<td>75 125 3.7%</td>
</tr>
<tr>
<td>• • • • XY</td>
<td>100 440 1.5%</td>
<td>22 169 0.6%</td>
<td>16 66 1.7%</td>
<td>50 104 4.4%</td>
<td>75 125 4.8%</td>
</tr>
<tr>
<td>• • • • X</td>
<td>25 110 6.6%</td>
<td>25 122 4.3%</td>
<td>25 99 5.6%</td>
<td>50 104 11.6%</td>
<td>75 125 20.0%</td>
</tr>
<tr>
<td>• • • • X</td>
<td>25 110 7.8%</td>
<td>22 169 3.8%</td>
<td>40 158 4.8%</td>
<td>33 69 17.8%</td>
<td>75 125 11.0%</td>
</tr>
<tr>
<td>• • • • Y</td>
<td>33 147 5.2%</td>
<td>16 81 2.3%</td>
<td>20 79 3.6%</td>
<td>20 42 10.9%</td>
<td>75 125 8.8%</td>
</tr>
<tr>
<td>• • • • Y</td>
<td>33 147 5.2%</td>
<td>16 81 2.3%</td>
<td>20 79 3.6%</td>
<td>20 42 10.9%</td>
<td>75 125 8.8%</td>
</tr>
<tr>
<td>• • • • P</td>
<td>28 140 3.8%</td>
<td>22 169 2.3%</td>
<td>25 99 1.8%</td>
<td>33 69 13.8%</td>
<td>75 125 25.0%</td>
</tr>
<tr>
<td>• • • • XY</td>
<td>33 147 3.8%</td>
<td>14 70 1.5%</td>
<td>20 79 2.8%</td>
<td>33 69 6.3%</td>
<td>75 125 5.6%</td>
</tr>
<tr>
<td>• • • • XY</td>
<td>33 147 4.5%</td>
<td>10 51 1.9%</td>
<td>28 113 2.5%</td>
<td>14 30 13.9%</td>
<td>75 125 4.7%</td>
</tr>
<tr>
<td>• • • • XY P</td>
<td>10 51 1.9%</td>
<td>28 113 1.8%</td>
<td>100 208 3.8%</td>
<td>75 125 2.6%</td>
<td></td>
</tr>
<tr>
<td>• • • • XY’</td>
<td>30 147 4.3%</td>
<td>25 122 3.2%</td>
<td>20 79 3.8%</td>
<td>14 30 19.0%</td>
<td>75 125 7.0%</td>
</tr>
<tr>
<td>• • • • X</td>
<td>30 147 6.3%</td>
<td>25 122 3.2%</td>
<td>20 79 3.8%</td>
<td>14 30 19.0%</td>
<td>75 125 7.0%</td>
</tr>
<tr>
<td>• • • • Y</td>
<td>50 220 3.1%</td>
<td>28 140 1.2%</td>
<td>100 496 1.3%</td>
<td>33 69 8.9%</td>
<td>75 125 6.2%</td>
</tr>
<tr>
<td>• • • • Y</td>
<td>50 220 3.8%</td>
<td>15 75 2.4%</td>
<td>22 88 0.9%</td>
<td>25 52 10.6%</td>
<td>75 125 4.5%</td>
</tr>
<tr>
<td>• • • • P</td>
<td>25 110 4.4%</td>
<td>40 196 1.6%</td>
<td>66 264 1.8%</td>
<td>50 104 6.5%</td>
<td>75 125 17.2%</td>
</tr>
<tr>
<td>• • • • P</td>
<td>20 88 5.9%</td>
<td>100 489 2.0%</td>
<td>40 158 1.7%</td>
<td>50 25 9.0%</td>
<td>75 125 17.7%</td>
</tr>
<tr>
<td>• • • • XY</td>
<td>100 440 1.6%</td>
<td>28 140 1.0%</td>
<td>66 264 1.6%</td>
<td>50 25 6.9%</td>
<td>75 125 3.8%</td>
</tr>
<tr>
<td>• • • • XY</td>
<td>100 440 2.3%</td>
<td>9 44 1.4%</td>
<td>66 264 1.6%</td>
<td>11 23 14.7%</td>
<td>37 63 3.3%</td>
</tr>
<tr>
<td>• • • • XY P</td>
<td>100 440 1.5%</td>
<td>200 978 0.4%</td>
<td>200 792 0.7%</td>
<td>25 52 6.9%</td>
<td>75 125 5.5%</td>
</tr>
<tr>
<td>• • • • XY P</td>
<td>100 440 1.8%</td>
<td>100 513 0.8%</td>
<td>25 99 0.3%</td>
<td>11 23 14.7%</td>
<td>75 125 4.2%</td>
</tr>
<tr>
<td>• • • • X</td>
<td>50 220 3.4%</td>
<td>50 244 2.7%</td>
<td>40 158 2.7%</td>
<td>100 208 7.3%</td>
<td>75 125 10.9%</td>
</tr>
<tr>
<td>• • • • X</td>
<td>30 220 4.1%</td>
<td>25 122 2.3%</td>
<td>40 158 3.3%</td>
<td>33 69 11.9%</td>
<td>75 125 7.3%</td>
</tr>
<tr>
<td>• • • • Y</td>
<td>25 110 2.6%</td>
<td>50 244 1.5%</td>
<td>66 264 1.7%</td>
<td>50 104 5.6%</td>
<td>75 125 8.2%</td>
</tr>
<tr>
<td>• • • • Y</td>
<td>25 110 3.1%</td>
<td>22 169 1.8%</td>
<td>50 198 1.2%</td>
<td>50 104 13.8%</td>
<td>75 125 7.2%</td>
</tr>
<tr>
<td>• • • • P</td>
<td>20 88 5.9%</td>
<td>28 140 2.1%</td>
<td>100 496 1.4%</td>
<td>50 104 6.1%</td>
<td>75 125 17.2%</td>
</tr>
<tr>
<td>• • • • P</td>
<td>20 88 5.9%</td>
<td>40 196 2.3%</td>
<td>66 264 0.9%</td>
<td>25 52 10.3%</td>
<td>75 125 14.9%</td>
</tr>
<tr>
<td>• • • • XY</td>
<td>100 440 1.8%</td>
<td>50 244 0.8%</td>
<td>66 264 1.2%</td>
<td>100 208 3.5%</td>
<td>37 63 6.0%</td>
</tr>
<tr>
<td>• • • • XY</td>
<td>100 440 2.1%</td>
<td>25 122 0.9%</td>
<td>22 88 0.8%</td>
<td>33 69 9.2%</td>
<td>75 125 3.7%</td>
</tr>
<tr>
<td>• • • • XY P</td>
<td>20 88 1.8%</td>
<td>100 489 0.4%</td>
<td>40 158 0.3%</td>
<td>33 69 1.6%</td>
<td>75 125 5.5%</td>
</tr>
<tr>
<td>• • • • XY P</td>
<td>100 440 1.8%</td>
<td>22 169 0.8%</td>
<td>18 72 0.1%</td>
<td>33 69 4.6%</td>
<td>75 125 5.0%</td>
</tr>
</tbody>
</table>

The results of the experiments showed that, in most cases, the accuracy started to increase until some point, and then it started to decrease. Figure 4.2 shows two
Chapter 4. Sampling frequency based online signature verification system

Figure 4.2: Expected (left) vs. typical (right) behavior of the EER as a function of sample points on SVC2004.

Table 4.2: Databases used and their statistics

<table>
<thead>
<tr>
<th>Database</th>
<th>Language</th>
<th>signers/total signatures</th>
<th>Initial average points</th>
<th>Device sampling rate</th>
<th>Best sampling rates</th>
<th>Best average points</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC2004</td>
<td>English/Chinese</td>
<td>40/1600</td>
<td>198</td>
<td>100 Hz</td>
<td>14 Hz – 50 Hz</td>
<td>30–104</td>
</tr>
<tr>
<td>MCYT-100</td>
<td>Spanish</td>
<td>100/5000</td>
<td>440</td>
<td>100 Hz</td>
<td>20 Hz – 50 Hz</td>
<td>88–220</td>
</tr>
<tr>
<td>SigComp’11</td>
<td>Dutch</td>
<td>64/1905</td>
<td>978</td>
<td>200 Hz</td>
<td>16 Hz – 50 Hz</td>
<td>81–244</td>
</tr>
<tr>
<td>SigComp’11</td>
<td>Chinese</td>
<td>20/1339</td>
<td>792</td>
<td>200 Hz</td>
<td>18 Hz – 50 Hz</td>
<td>72–198</td>
</tr>
<tr>
<td>SigWiComp’15</td>
<td>German</td>
<td>30/750</td>
<td>125</td>
<td>75 Hz</td>
<td>37 Hz – 75 Hz</td>
<td>63–125</td>
</tr>
</tbody>
</table>

examples for both cases of the effect of the average number of sample points on the EER from SVC2004. In fact, it follows the second behavior in 88.1% of the configurations of all databases acquired between 100 Hz and 200 Hz, 97.5% for SigComp’11 (Dutch), in 92.5% for SigComp’11 (Chinese), 87.5% for SVC-2004, and 75% for MCYT-100; only a few combinations provided better results when using the initial sampling rate. Furthermore, in these databases, 85.6% of the best results were obtained using a sampling frequency of less than or equal to 50 Hz.

In the case of SigWiComp’15, the database was acquired using a relatively low frequency of 75 Hz, thus downsampling the good results, which also means that, for all the databases, 88.5% of the best results of the configuration were obtained using a sampling frequency of less than or equal to 75 Hz, and 70% when using a sampling frequency of less than or equal to 50 Hz. Figure 4.3 shows some examples of the effect of sampling frequency on the EER.
Figure 4.3: Examples from other databases of the effect of sampling frequency on the EER.

4.1.5.2 The best frequency ranges

My observations showed that, in most cases, the accuracy increased until some point and then started decreasing or stagnating. However, I also observed that there was a specific range where we could obtain the best results. As an example, Table 4.1 shows that, for the SigComp’11 (Dutch) database, 93% of the best results were obtained when using an average sampling frequency of less than or equal to 50 Hz. Similar observations were made with the other databases. In general, we can say that the majority of the best results were obtained using a range of around 15 Hz to 50 Hz. These ranges are shown in Figure 4.4.

4.1.5.3 The best sample count ranges

The effect of sample count followed the same behavior as that of the frequency rate as they are related. Table 4.2 shows the best results of the sample counts.
Chapter 4. Sampling frequency based online signature verification system

Figure 4.4: Sampling frequencies of the best results for all the tests.

Figure 4.5: Sample counts of the best results for all the tests.

We can see that the sample count ranges between 30 and 104 for SVC2004 and between around 60 and 240 for the other databases that provided the lowest error rate. Figure 4.5 shows the best sample counts for all the tests.

4.1.5.4 Sampling restrictions

The results showed that there are three cases for the frequency range. The first one is where the frequency rate is low or under a specific range (under-sampling). In this case, the error rate is high because it does not provide sufficient information.
about the signature that makes it unique and allows it to be distinguished from other signatures. In the second case, the best results can be obtained when the sampling rate and the signature points are in a specific range where they are neither low nor high. This makes sense because it was shown that the maximum frequency band limit for online signatures is 20 Hz – 30 Hz and that a Nyquist rate of 40 Hz – 60 Hz will be sufficient to provide adequate information about the signal without any redundant information. The third case is where the frequency is above a specific range where the accuracy of the results decreases again. I believe that the redundant data not only may provide redundant information that will not help in obtaining better results but may also worsen it (over-sampling). These cases are related to the situation with analog signal sampling, where choosing the wrong sampling frequency may produce under-sampling or oversampling issues.

4.1.5.5 Comparison

Although this study aimed to measure the effect of the sampling frequency on the accuracy, it is also worth mentioning that some of the verification systems applied here achieved competitive results to state-of-the-art systems. In Table 4.3 I show some of the best results achieved with down-sampling compared to other results for different databases.

4.2 Signer-dependent sampling frequency approach for signature verification

In the previous section, I showed that choosing a different sample rate or signature points count provides better accuracy in online signature verification systems. However, the sampling rate that minimizes the error may vary between the databases and signer. In this section, I focus on the results of using signer-dependent sampling frequency rather than using the same sampling frequency for the entire database.

A total of 500 tests were applied in this work using several online signature verification systems to assure the quality of the results. The signatures were down-
Chapter 4. Sampling frequency based online signature verification system

Table 4.3: A comparison with recent similar results in the field

<table>
<thead>
<tr>
<th>Database</th>
<th>EER%</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCYT-100</td>
<td>3.69</td>
<td>Xia et al. [Xia et al., 2018]</td>
</tr>
<tr>
<td>MCYT-100</td>
<td>3.05</td>
<td>Sharma et al. [Sharma and Sundaram, 2016c]</td>
</tr>
<tr>
<td>MCYT-100</td>
<td>2.76</td>
<td>Sharma et al. [Sharma and Sundaram, 2018]</td>
</tr>
<tr>
<td>MCYT-100</td>
<td>1.81</td>
<td>Lai et al. [Lai and Jin, 2018]</td>
</tr>
<tr>
<td><strong>MCYT-100</strong></td>
<td><strong>1.8</strong></td>
<td><strong>Proposed</strong></td>
</tr>
<tr>
<td>MCYT-100</td>
<td>1.34</td>
<td>Okawa et al. [Okawa, 2019]</td>
</tr>
<tr>
<td>MCYT-100</td>
<td>1.28</td>
<td>Okawa et al. [Okawa, 2020]</td>
</tr>
<tr>
<td>SVC2004</td>
<td>6.65</td>
<td>Wang et al. [Wang et al., 2011]</td>
</tr>
<tr>
<td>SVC2004</td>
<td>3.41</td>
<td>Alpar et al. [Alpar, 2018]</td>
</tr>
<tr>
<td>SVC2004</td>
<td>2.73</td>
<td>Sharma et al. [Sharma and Sundaram, 2016b]</td>
</tr>
<tr>
<td>SVC2004</td>
<td>1.8</td>
<td>Foroozandeh et al. [Foroozandeh et al., 2020]</td>
</tr>
<tr>
<td><strong>SVC2004</strong></td>
<td><strong>1.6</strong></td>
<td><strong>Proposed</strong></td>
</tr>
<tr>
<td>SVC2004</td>
<td>0.83</td>
<td>Fayyaz et al. [Fayyaz et al., 2015a]</td>
</tr>
<tr>
<td>SigComp’11 (Dutch)</td>
<td>3.73</td>
<td>[Liwicki et al., 2011]</td>
</tr>
<tr>
<td>SigComp’11 (Dutch)</td>
<td>5.5</td>
<td>[Parodi and Gómez, 2014]</td>
</tr>
<tr>
<td>SigComp’11 (Dutch)</td>
<td>3.65</td>
<td>[Liwicki et al., 2011]</td>
</tr>
<tr>
<td><strong>SigComp’11 (Dutch)</strong></td>
<td><strong>0.6</strong></td>
<td><strong>Proposed</strong></td>
</tr>
<tr>
<td>SigComp’11 (Chinese)</td>
<td>14.68</td>
<td>[Liwicki et al., 2011]</td>
</tr>
<tr>
<td>SigComp’11 (Chinese)</td>
<td>8.93</td>
<td>[Parodi and Gómez, 2014]</td>
</tr>
<tr>
<td>SigComp’11 (Chinese)</td>
<td>6.83</td>
<td>[Liwicki et al., 2011]</td>
</tr>
<tr>
<td><strong>SigComp’11 (Chinese)</strong></td>
<td><strong>0.1</strong></td>
<td><strong>Proposed</strong></td>
</tr>
</tbody>
</table>

sampled and tested for different sample rates in each verification test. The best sampling frequency for each signer was assigned in a testing verification system using only the reference signatures (provided by the signer) before using them later in the verification process.

4.2.1 Methodology

The proposed technique is based on choosing the best sampling frequency for each signer before starting the verification process. In real-life situations, the signer provides some signatures as references. These signatures are used to compare the tested signature and check if it is forged or genuine. Therefore, the proposed technique will use only these references for choosing the signer optimal sampling frequency to evaluate the system in a real-life situation where no more genuine or forged signatures are available. The references were divided into training and testing sets, and only the false rejection rate (FRR) was calculated and used to
choose the best sampling frequency as only genuine signatures are available at this point. Nevertheless, to evaluate the efficiency of the proposed technique, more genuine (not used in the previous step) and forged signatures are used later to calculate the error rate in the evaluation step. Although the first step contains all the main stages of a signature verification process, it is used as a preprocessing step to improve the entire system’s quality later. The same verifier used in the first step, where we choose the best sampling frequency for each signer, is used in the system evaluation step to obtain the actual improvement under the same circumstances. Several inputs for each stage of the online signature verification process were used to obtain more results that help better understanding the effect of using a signer-dependent sampling frequency.

Five different databases were used for data acquisition, the SVC2004, the MCYT-100, the Dutch and Chinese subsets of the SigComp’11 and the SigComp’13 database. In this experiment, scaling, translation, z-normalization, and their combinations were used (see Section 4.1.2. Five combinations of features are used as follows: X, Y, P, XY, XYP. These choices are made based on the main conclusions of the previous chapter.

In each test for each database, preprocessing methods are chosen, with selected features in the verifier for the current test. As mentioned before, a DTW-based verifier is used.

### 4.2.1.1 Proposed technique

The previous steps were combined to provide many tests. Usually, the signer provides 10 or more reference signatures. Therefore, in the proposed system I assumed that each signer has 10 signatures, which as discussed before are later divided in the training phase. As a result, the combinations tested using 3-7 samples \((n)\) as references, 10-n for testing. The values of \(n=1\) and \(n=2\) were neglected as they will not provide enough information that can be considered for the calculations, and no more than 10 signatures in total are used at this step. In Figure 4.6, the combinations are shown.

After choosing the current verifier, it is first applied using different sampling rates. In each iteration, the sampling rate is changed, and the AER is calculated for
Figure 4.6: The combinations of the tests applied in the experiments.

Each signer. The number of iterations depends on the initial sampling frequency, 20-40 iterations for 100 Hz - 200 Hz databases, and 100 iterations if the signature data is very large. The number of iterations here reflects the amount of data signatures have in each database, this insures that almost every range of signature sample points is tested.

In each iteration, the sampling frequency is changed, and the system is applied using it. Since $n$ number of samples are used as references to calculate the threshold in the verification process, the rest of the signatures $(10-n)$ are used to test the accuracy when using the current sampling frequency. Only references are used in this phase, so only FRR of the $(10-n)$ signatures are used to compare the results and assign the best sampling frequency to each signer.

After assigning each signer with his/her best sampling frequency, the same verification system with the same preprocessing methods, features, and classification algorithm are applied to each signer individually, and all error rates are calculated for them. Later the average error rate overall the signers is calculated. In order to evaluate the current verification system and its efficiency when using signer-dependent frequencies, we need to apply the same verification system for the same database but using the initial sampling frequency and then compare the results of both methods and calculate the accuracy improvement. Figure 4.7 provides a brief description of the work process.
4.2.2 Results and evaluation

A comparison study has been conducted to check whether the proposed method has improved the existing online signature verification system. Since sometimes other factors may affect the accuracy of the results, several tests will provide a more accurate evaluation by investigating the behavior of the majority of these systems to avoid any other factors that may impact the results.

It also helps in choosing the best or optimal methods that can be used to provide the most accurate verification systems. In this section, all cases are discussed and evaluated to show the improvement in both randomly chosen systems and optimal systems.

4.2.2.1 All combinations evaluation

Testing all the scenarios by using scaling, translation, and z-normalization for preprocessing and using \( n \): 3-7 (500 tests) showed that the accuracy improved in 72% of the tests. The improvement in the accuracy reached up to 8.38%. However, 20 tests showed a decrease in the accuracy, and eight tests resulted in the same accuracy. Overall, 80% of the tests provided better or equal verification accuracy regardless of the preprocessing techniques applied or features selected or
any number of samples used. Figure 4.8 shows the accuracy improvement for all cases.

![Accuracy improvement (all tests)](image)

Figure 4.8: Percentage of experiments (vertical axis) where the accuracy improved when using the signer-dependant sampling frequency approach (all the tests).

### 4.2.2.2 Preprocessing methods effect

Preprocessing methods have a significant impact on the verification accuracy. The experiment showed that $z$-normalization provides the most accurate results. Therefore, the results of the test where $z$-normalization was used showed accuracy improvement in 88% of the cases, there was no change in 1% of the cases, and the accuracy decreased in 11% of the cases, see Figure 4.9.

### 4.2.2.3 The optimal system

In the previous subsection, I showed that choosing the preprocessing method will affect the accuracy. Also, it is significant to select the samples number. Although $n=3:7$ was applied, the best results were achieved when using 3 or 6 samples. Combining these facts into one verification system will provide the most accurate system. Choosing six samples as references and $z$-normalization for preprocessing while using five different databases and five different features (25 tests) have led to only one negative result, one result with no change and 23 results (92%) with accuracy improvement up to around 8.4%, see Figure 4.10.
Chapter 4. Sampling frequency based online signature verification system

Figure 4.9: Percentage of experiments (vertical axis) where the accuracy improved when using the signer-dependant sampling frequency approach (using $z$-normalization).

Figure 4.10: Percentage of experiments (vertical axis) where the accuracy improved when using the signer-dependant sampling frequency approach (using $z$-normalization and 6 samples).

4.3 Chapter summary

In this thesis, I studied the effect of the sampling rate of the input devices used for signature acquisition and the number of sample points on the accuracy of online signature verification systems. Several configurations of a DTW-based verification system were used to assess the achievable EER at different sampling rates. Altogether, I conducted 5600 different experiments, which helped generalize the results.
regardless of the effect of other factors that may affect the system’s accuracy. To my knowledge, these properties have never been studied within the scope of online signature verification.

The results showed that the majority of the best results could be obtained using a sampling frequency between 15 Hz and 50 Hz and a sample count between 60 and 240 points. Using frequencies lower than these ranges significantly decreased the accuracy, whereas using higher frequencies decreased or did not affect the accuracy in 88.1% of the configurations of all databases acquired between 100 Hz and 200 Hz, 97.5% for SigComp’11 (Dutch), in 92.5% for SigComp’11 (Chinese), 87.5% for SVC-2004, and 75% for the MCYT-100. For these databases, 85.6% of the best results were obtained used a sampling frequency of less than or equal to 50 Hz. The best results of SigWiComp’15 were obtained using a lower sampling frequency (75 Hz), which led to better accuracy even for the initial value. Moreover, the results showed that, for all the databases, 88.5% of the best results were obtained using a sampling frequency of less than or equal to 75 Hz, and 70% when using a sampling frequency of less than or equal to 50 Hz.

As the sampling rate and the sample count are strongly correlated, it is too early to conclude which one of the two plays more significant role in the observed relation; therefore, I presented my results by including both. Regardless of this fact, I can state that, in classic DTW-based signature verification, using sampling frequencies higher than 100 Hz will not improve the accuracy of the systems but will instead increase the computational cost of the verification.

The second part of the chapter showed that using signer-dependent custom sampling frequency can improve the accuracy of the online signature verification systems, especially when choosing the optimal verifier. Five hundred different tests were used to avoid any other factors that may affect the results, such as the database used, the preprocessing method applied, or features selected. Also, different samples were used to evaluate the proposed technique and obtain the best method for assigning the signer-dependent sampling frequency.

The results showed that in 80% of the 500 tests, the accuracy improved or at least did not change (72% improved). Also, the ratio of improved results reached 92% when chosen the optimal preprocessing methods and number of samples.
Chapter 5

K-nearest neighbor algorithm for online signature verification

This chapter discusses three main contributions related to the $k$-nearest neighbor ($k$-NN) algorithm in online signature verification. First, the main parameters of the $j_{k}$-nearest neighbor ($j_{k}$-NN) algorithm are evaluated and analyzed. Then I propose an optimized $j_{k}$-NN algorithm for online signature verification. Furthermore, a combination of the $k$-NN and dynamic time warping (DTW) algorithms are used to build a competitive verification system.

5.1 Introduction

Besides being used for multi-class classification, the nearest neighbor is one of the available methods used to solve the one-class classification problem [Manevitz and Yousef, 2001][Cabral et al., 2009].

$k$-NN is a non-parametric classification approach proposed by Thomas Cover [Cover and Hart, 1967]. In the one class $k$-NN algorithm, the $k$ nearest neighbors of the first nearest neighbor classify the tested object. Each distance $d$ between the tested signature ($S$) nearest neighbor ($S_{nn}$) and its ($S_{knn}$) nearest neighbor is measured, and the calculated average of the latter ($D_{avg}$). If the distance between the tested signature and its nearest neighbor is less than or equal to the threshold
$\theta * D_{avg}$ then $S$ is classified as genuine otherwise forged:

$$d (S, S_{nn}) < \theta \frac{1}{K} \sum_{k=1}^{K} d(S_{mn}, S_{knn})$$

(5.1)

In the one class $k$-NN approach, only the distance between the tested signature and its first nearest neighbor verifies the signature to be genuine or forged. It compares the average distances between the first nearest neighbor and its $k$ nearest neighbors. While in the $jk$-NN method, the nearest neighbors $j$ are used for the classification by comparing each $j$th neighbor with its own $k$ nearest neighbors individually, then use these calculation to classify the signature as genuine or forged, see Figure 5.1. We can say that the $k$-NN and the $jk$-NN are the same if $j = 1$. Each neighbor is tested using the $k$-NN algorithm. If the majority are accepted, then the signature is classified as genuine, otherwise forged.

![Figure 5.1: k-NN (Left) and jk-NN (Right).](image)

The $jk$-NN algorithm can be formalized as the following:

$$\sum_{j=1}^{J} \left[ d (S_j, S_{jnn}) < \theta \frac{1}{K} \sum_{k=1}^{K} d(S_{jnn}, S_{jknn}) \right] > 0.5$$

(5.2)
5.2 Related work

Khan et al. [Khan and Ahmad, 2018] presented a theoretical analysis of $k$-NN one-class classification algorithm variants. Their results showed that by optimizing the parameters, $jk$-NN could achieve good results, especially with low values of $j$ and high values of $k$. Their evaluation tested several datasets but not signature datasets. Pippin [Pippin, 2004] suggested a technique for online authentication of signatures. It extracts global signature features and compares these features to stored signature models using the $k$-NN classification.

Yang et al. [Yang et al., 2018] suggested a new writer-dependent online verification technique for signature verification based on Relief and using $k$-NN for classification. They used the SVC2004 database in their work and achieved an average error rate of 5.312%. Nanni [Nanni, 2006] used $k$-NN for one class online signature verification system and achieved a 12.2% error rate and 6.3% when using 5 and 20 skilled signatures for the training set on the MCYT-100 database, respectively.

$k$-NN was also used for offline signature verification systems [Vargas et al., 2008, Harfiya et al., 2017, Vickram and Swapna, 2016]. Azmi et al. [Azmi et al., 2017] used $k$-NN and Freeman chain code (FCC) in their work and achieved 9.85% AER on the MCYT-100 database. Abdelrahaman et al. [Abdelrahaman and Abdallah, 2013] also achieved 80% accuracy by using $k$-NN for their offline SVS. $k$-NN is also applied in related fields such as text recognition [Jo et al., 2015], iris recognition [Winston and Hemanth, 2020], and emotion detection [Saxena et al., 2020].

5.3 Evaluation of the algorithm parameters’ effect

To build a high accuracy $jk$-NN-based online signature verification system, the main parameters that affect the accuracy need to be evaluated and analyzed. For this purpose, several main parameters are used and applied on different databases such as the SVC2004, MCYT-100, SigComp’11 (Dutch), Sig-
WiComp2015 databases. The details of the used database are presented in Table 2.1. In addition, five combinations of $X$, $Y$, $P$, $XY$, and $XYP$ were used as features, and the z-normalization algorithm was used in the preprocessing step.

The $jk$-NN classifier was implemented and used for the classification after applying the preprocessing algorithm to the acquired signatures and extracting the required features. Each signature is compared to its $j$ nearest neighbors and their $k$ nearest neighbors ($k$-nearest neighbors of each $j$) and then classified as genuine or forged based on the computed thresholds.

The experiment started by using various reference numbers and thresholds, evaluating and analyzing the verification accuracy’s performance. After that, I examined several $j$ and $k$ values and the effect of adjusting the number of nearest neighbors upon the results and compared the $k$-NN and $jk$-NN algorithms. For the $jk$-NN online signature verifier, the $j$ values, $k$ values, number of references used, and threshold selection (value of $\theta$) are the significant factors determining the performance. In the following subsections, each of these factors is discussed and evaluated.

5.3.1 Number of references and threshold scale

The tested signature references number and threshold scaling ranges were 7-17 and 0.8-1.8, respectively. These ranges were chosen based on an initial experiment where fewer or more values were extremely biased and had no useful information to the test. The experiments showed greater accuracy when using more than ten original signatures as references, particularly between 13 and 15 (see Table 5.1). A larger number of reference signatures can provide a better representation of intra-class variations. It is worth mentioning that 13-15 references are high numbers considering a real-life scenario where usually less than this range of reference signatures are available. However, at this point, these values are included to study the general effect of the number of references.

To accept a test signature, the average distance between it and the $j$ nearest signatures should be less than the average of the average distance between each $j$ signature and its $k$-NN signatures or a predetermined scale of that threshold. As shown in Table 5.1, in the top 10 results of each database, the threshold ($\theta$) used
Table 5.1: The best results of the experiments.

<table>
<thead>
<tr>
<th>References</th>
<th>Threshold</th>
<th>Min AER%</th>
<th>References</th>
<th>Threshold</th>
<th>Min AER%</th>
<th>References</th>
<th>Threshold</th>
<th>Min AER%</th>
<th>References</th>
<th>Threshold</th>
<th>Min AER%</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>1.2</td>
<td>3.938</td>
<td>13</td>
<td>1.6</td>
<td>2.605</td>
<td>15</td>
<td>1.5</td>
<td>1.752</td>
<td>13</td>
<td>1.5</td>
<td>6.000</td>
</tr>
<tr>
<td>15</td>
<td>1.4</td>
<td>4.063</td>
<td>13</td>
<td>1.7</td>
<td>2.678</td>
<td>13</td>
<td>1.5</td>
<td>1.841</td>
<td>13</td>
<td>1.3</td>
<td>7.000</td>
</tr>
<tr>
<td>15</td>
<td>1.5</td>
<td>4.063</td>
<td>13</td>
<td>1.8</td>
<td>2.690</td>
<td>14</td>
<td>1.5</td>
<td>1.917</td>
<td>13</td>
<td>1.2</td>
<td>7.167</td>
</tr>
<tr>
<td>15</td>
<td>1.6</td>
<td>4.125</td>
<td>12</td>
<td>1.8</td>
<td>2.691</td>
<td>13</td>
<td>1.8</td>
<td>1.929</td>
<td>13</td>
<td>1.4</td>
<td>7.167</td>
</tr>
<tr>
<td>15</td>
<td>1.3</td>
<td>4.188</td>
<td>12</td>
<td>1.6</td>
<td>2.717</td>
<td>13</td>
<td>1.6</td>
<td>1.978</td>
<td>11</td>
<td>1.3</td>
<td>7.250</td>
</tr>
<tr>
<td>15</td>
<td>1.1</td>
<td>4.438</td>
<td>14</td>
<td>1.6</td>
<td>2.722</td>
<td>13</td>
<td>1.7</td>
<td>1.978</td>
<td>11</td>
<td>1.5</td>
<td>7.500</td>
</tr>
<tr>
<td>15</td>
<td>1.7</td>
<td>4.438</td>
<td>13</td>
<td>1.5</td>
<td>2.738</td>
<td>15</td>
<td>1.6</td>
<td>2.007</td>
<td>10</td>
<td>1.4</td>
<td>7.667</td>
</tr>
<tr>
<td>14</td>
<td>1.5</td>
<td>4.729</td>
<td>12</td>
<td>1.5</td>
<td>2.742</td>
<td>11</td>
<td>1.6</td>
<td>2.036</td>
<td>12</td>
<td>1.3</td>
<td>7.667</td>
</tr>
<tr>
<td>14</td>
<td>1.2</td>
<td>4.750</td>
<td>14</td>
<td>1.5</td>
<td>2.742</td>
<td>11</td>
<td>1.7</td>
<td>2.057</td>
<td>12</td>
<td>1.4</td>
<td>7.778</td>
</tr>
<tr>
<td>14</td>
<td>1.4</td>
<td>4.750</td>
<td>15</td>
<td>1.7</td>
<td>2.760</td>
<td>13</td>
<td>1.4</td>
<td>2.064</td>
<td>12</td>
<td>1.5</td>
<td>7.778</td>
</tr>
</tbody>
</table>

were between 1.1–1.8. The tested threshold were between 0.8–1.8, but using values below 1 showed less accurate results. We can say that the optimal value of $\theta$ is between 1.2–1.6. A combination of 15 references and a scaling threshold of $\theta = 1.5$ will perform very well in a $jk$-NN-based online signature verification system.

### 5.3.2 $j$ and $k$ values

The $j$ and $k$ values have a different effect on FAR and FRR. FRR improves with small values of $j$, and larger values of $k$. This effect occurs because of the more distant neighbors included with increasing $k$, which facilitates the chosen threshold and accepts more signatures, thereby decreasing the rejection rate. Also, this has the exact opposite effect on FAR for the same reason; more signatures are accepted in this situation, leading to the acceptance of some forged signatures, which leads to a higher FAR. Thus, FAR is smaller with smaller $k$ and $j$’s larger values (see Table 5.2).

These two effects require consideration when choosing $j$ and $k$’s value since AER is affected by FAR and FRR. In this approach, the parameters’ optimal values will
be the values that tolerate both error types. The previous test showed that 15 signatures as references are ideal for this experiments. Therefore, the values of the tested $j$ and $k$ were 1-15.

### 5.3.3 $j^k$-NN performance

A $j^k$-NN classifier is used for verification instead of $k$-NN for the proposed algorithm. Although $k$-NN provides good accuracy, the results have shown that using $j^k$-NN can improve the verification system’s accuracy compared to the $k$-NN algorithm. A comparison between the two algorithms is shown in Figure 5.2. For the SVC2004 database, the increase in accuracy was 2.02%, 0.59% for the MCYT-100 database, 0.73% for the SigComp’11 database, and 10% for the SigComp’15 database.

![Accuracy Improvement using $j^k$-NN](image)

Figure 5.2: Accuracy improvement of the $j^k$-NN algorithm comparing to the $k$-NN algorithm.

Figure 5.2 shows that the $j^k$-NN algorithm achieved good performance for the different databases, a 3.93% error rate when using the SVC2004 database, 2.6% for MCYT-100, 1.75% for SigComp’11, and 6% for the SigWiComp’15 database. However, this approach would not always be feasible in practice, so I suggested an
Table 5.2: The effect of $j$ (up to down) and $k$ (left to right) on FAR (up) and FRR (bottom).

<table>
<thead>
<tr>
<th>FAR</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0125</td>
<td>0.0125</td>
<td>0.00625</td>
<td>0.011</td>
<td>0.006375</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.01125</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
</tr>
<tr>
<td>1</td>
<td>0.0125</td>
<td>0.0125</td>
<td>0.00625</td>
<td>0.011</td>
<td>0.006375</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.01125</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
</tr>
<tr>
<td>2</td>
<td>0.0125</td>
<td>0.0125</td>
<td>0.00625</td>
<td>0.011</td>
<td>0.006375</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.01125</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
</tr>
<tr>
<td>3</td>
<td>0.0125</td>
<td>0.0125</td>
<td>0.00625</td>
<td>0.011</td>
<td>0.006375</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.01125</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
</tr>
<tr>
<td>4</td>
<td>0.0125</td>
<td>0.0125</td>
<td>0.00625</td>
<td>0.011</td>
<td>0.006375</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.01125</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
</tr>
<tr>
<td>5</td>
<td>0.0125</td>
<td>0.0125</td>
<td>0.00625</td>
<td>0.011</td>
<td>0.006375</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.01125</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FRR</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0125</td>
<td>0.0125</td>
<td>0.00625</td>
<td>0.011</td>
<td>0.006375</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.01125</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
</tr>
<tr>
<td>1.0125</td>
<td>0.0125</td>
<td>0.00625</td>
<td>0.011</td>
<td>0.006375</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.01125</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
</tr>
<tr>
<td>1.0125</td>
<td>0.0125</td>
<td>0.00625</td>
<td>0.011</td>
<td>0.006375</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.01125</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
</tr>
<tr>
<td>1.0125</td>
<td>0.0125</td>
<td>0.00625</td>
<td>0.011</td>
<td>0.006375</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.01125</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
</tr>
<tr>
<td>1.0125</td>
<td>0.0125</td>
<td>0.00625</td>
<td>0.011</td>
<td>0.006375</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.01125</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
<td>0.00625</td>
</tr>
</tbody>
</table>
improved \(jk\)-NN in the next section using the previous evaluation of the parameters and optimizing the \(k\) value using only available references.

## 5.4 Optimized \(jk\)-NN based online signature verification

In the previous subsections, the minimal AER achieved in each experiment was presented and discussed. I obtained minimal AER by selecting the best values of \(j\) and \(k\), which cannot be reliably predetermined in real circumstances. Nevertheless, I used the result for the SVS after examining the effect of the reference number and the best threshold scale, which can provide good results in most situations.

This section introduces an algorithm based on the minimum value of FRR reached within the training set to choose the best value of \(k\), calculated in real-life circumstances where a certain number of signature references are available and used for this purpose. Using the previous evaluation of the algorithm parameters, the values \(j=5\) and \(\theta =1.5\) are used with 15 reference signatures in the proposed algorithm. The idea is to divide the references (\(R\)) into two groups, the first group (\(R_t\)) is used to calculate the threshold, while the other group is used for testing. FRR is calculated in each iteration using different values of \(k\) from the (\(K_s\)) group of values. The best value is assigned for \(k\) (\(K_{opt}\)) to the \(k\) that provided the minimum \(FRR(K)\) (FRR using the \(k\)-NN) among all \(K_s\). The SVS can use these values in the verification phase. The minimum value of (FRR) will not always provide the optimal value of \((k)\) since the references \(R\), and \(R_t\) are not the same and will provide different results. Still, it will indicate one of the best values of \(k\) that can produce a very accurate result. The new formula of the algorithm is presented in the following equations:

\[
j \in J^* \circledast: \left[ d \left(S_j, S_{j_{nn}}\right) < \theta \frac{1}{K_{opt}} \sum_{k=1}^{K_{opt}-1} d \left(S_{j_{nn}}, S_{j_{knn}}\right) \right] > 0.5 \tag{5.3}
\]

Where

\[
J^* \in \{1, 2, 3, 4, 5\} \tag{5.4}
\]
and $K_{opt}$ is the value of $k$ that provides:

$$\min_{\forall K \in K_s} FRR(K)$$  \hfill (5.5)

From the previous conclusions, I chose $R_t$ to be 10, $j = 5$, $\theta = 1.5$. The proposed algorithm chooses the best $k$ for each database and applies the SVS. The achieved accuracies are shown in Table 5.3. The accuracy of the approach proposed is encouraging. The achieved AER was 8% for SVC2004, 3.26% for MCYT-100, 13% for SigComp’15, and 2.22% for SigComp’11.

<table>
<thead>
<tr>
<th>Database</th>
<th>min FRR</th>
<th>AER</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC2004</td>
<td>5.83%</td>
<td>8.00%</td>
</tr>
<tr>
<td>MCYT100</td>
<td>2.67%</td>
<td>3.26%</td>
</tr>
<tr>
<td>SigComp’15</td>
<td>0.00%</td>
<td>13.00%</td>
</tr>
<tr>
<td>SigComp’11</td>
<td>2.60%</td>
<td>2.22%</td>
</tr>
</tbody>
</table>

### 5.5 $k$-NN and DTW for mobile scenario online signature verification

Some verification and distance measurement methods were introduced and evaluated in the previous chapters, such as DTW and $k$-NN. These two methods are commonly used in many verification and recognition systems. In this section, I use a novel combination of both methods to build an online signature verification system. The system was built to target the signature verification competition 2021 (SVC2021) [Tolosana et al., 2021b].

#### 5.5.1 Methodology

In the proposed system, $XYP$ features combination is used. First, the signature points were filtered by removing the points with zero or shallow pressure values except for the finger input signatures, as it has zero pressure values already for all
Chapter 5. K-nearest neighbor algorithm for online signature verification

the points. Then, \(X\), \(Y\), and \(P\) were scaled to the [0,1]. Each signature's maximum and minimum values were used to re-scale each feature to the mentioned range. For \(P\), the minimum value is always assumed to be 0. Thus, \(P\) values were not shifted. Both \(X\) and \(Y\) were shifted to the origin.

The system considers local thresholds to classify signatures and uses DTW and \(k\)-NN algorithms in the classification phase. DTW was used as a distance measurement between the signatures. The \(k\)-NN algorithm was used to select the reference signatures and calculate the upper and lower threshold, which plays a significant part in calculating the prediction of the tested signature.

The evaluation subset of the DeepSignDB (442 users) was used to test the system where the main parameters were tuned to select the optimal thresholds.

The distance was used to calculate the prediction \(P_q\) for the questioned signature \(S_q\) using a calculated forgery threshold \(F_{th}\), genuine threshold \(G_{th}\) and a scaling parameter \(s\) as following:

\[
P_q = \frac{s \ast F_{th} - d}{s \ast F_{th} - G_{th}}
\]

The prediction values are between zero and one, where zero represents a genuine signature, while one represents a forgery signature. Furthermore, a threshold is assigned to classify the signature as genuine or forged based on its prediction value.

5.5.2 SVC2021 & SVC-onGoing competitions

Signature Verification Competition 2021 (SVC2021) was established to evaluate the online verification systems using one approach for all and using a large-scale database. The competition consists of three tasks of three popular scenarios.

**Task1**: Analysis signatures that were acquired using a stylus as input (office scenario).

**Task2**: Analysis signatures that were acquired using a finger as input (mobile scenario).

**Task3**: Analysis of both office and mobile scenarios.

The DeepSignDB and the SVC2021_EvalDB databases were considered for the competition. Both databases have been discussed in detail in Section 2.4.2. The
SVC-onGoing competition is an extension of the SVC2021 competition, where participant can participate anytime.

These different scenarios and the huge database with numerous signatures acquired using different devices, ages, sessions, and different skill levels of forgery provide an excellent opportunity for researchers to test and compare their systems on a realistic benchmarking platform.

The competition consists of two stages development and evaluation.

In the development stage, only the DeepSignDB was provided to train the participants’ system. This provided testing under similar conditions and circumstances. The database was divided into training and evaluating sets. At this stage, participants had the chance to test and compare several versions of their system before the final evaluation.

In the evaluation stage, the novel SVC2021.EvalDB was considered. Besides the database, the organizers provided comparisons files.

In the development phase, the comparisons were provided using two strategies, 1vs1, where only one signature is available as a reference, and 4vs1, where four reference signatures are available. In the evaluation phase, only 1vs1 comparisons were available. The proposed system showed strong performance, especially when using 4vs1 comparisons.

5.5.3 Experimental results

Our team (SigStat team) participated in both stages of the competitions. Following are the results for both stages:

5.5.3.1 DeepSignDB

In the first stage of the competition, the DeepSignDB was available for the participants through the SVC onGoing platform. A baseline system based on DTW was presented by the competition creators [Tolosana et al., 2021b]. Some systems outperformed this baseline including our system. Our team ranked second overall the tasks in this stage. However, the proposed system achieved first position (gold medal) in Task2 with 5.81% EER. The system achieved 7.74% in Task1, while
ranked 3rd (brass medal) in Task3 with 7.71\%. Table 5.4 shows the results of the mobile scenario task.

Table 5.4: SVC2021-DeepSignDB - Task2 results.

<table>
<thead>
<tr>
<th>Position</th>
<th>Team name</th>
<th>EER (%)</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SigStat</td>
<td>5.81</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>DLCV Lab</td>
<td>6.58</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>SIG-Team</td>
<td>9.43</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Baseline DTW</td>
<td>10.16</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>TUSUR KIBEVS</td>
<td>12.68</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>JAIRG</td>
<td>12.86</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.5: SVC2021- SVC2021 EvalDB - Task2 results.

<table>
<thead>
<tr>
<th>Position</th>
<th>Team name</th>
<th>EER (%)</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DLCV Lab</td>
<td>7.4055</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>SIG-Team</td>
<td>10.1366</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>SigStat</td>
<td>13.2878</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>TUSUR KIBEVS</td>
<td>13.3929</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>MaD Lab</td>
<td>17.2269</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>JAIRG</td>
<td>18.4349</td>
<td>0</td>
</tr>
</tbody>
</table>

5.5.3.2 SVC2021 EvalDB

In the second stage of the competition, only 1vs1 comparisons were considered. Thus, the system works a bit differently from the development phase. However, the system still achieved a bronze medal when applying it to the mobile scenario tasks. Table 5.5 shows the competition results for Task2 (Mobile scenario); our system achieved 13.28\% EER (SigStat team).

The system achieved fourth place overall in this stage. The final results of all tasks for both stages are shown in Table 5.6.

As the organizers decided to keep it as an on-going competition where researchers can submit their systems and compare their results, the system is currently under improvement to achieve higher ranks in the competition.
Table 5.6: SigStat team results.

<table>
<thead>
<tr>
<th>Task</th>
<th>DeepSignDB</th>
<th>SVC_EvalDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task1</td>
<td>7.74%</td>
<td>11.75%</td>
</tr>
<tr>
<td>Task2</td>
<td>5.81%</td>
<td>13.29%</td>
</tr>
<tr>
<td>Task3</td>
<td>7.71%</td>
<td>14.48%</td>
</tr>
</tbody>
</table>

5.6 Chapter summary

A $jk$-NN-based online signature verification system is proposed in this thesis which is a generalized case of the $k$-NN algorithm. I began by evaluating the impact of the proposed algorithm’s main parameters, then presented and evaluated the verification method. A comparison was also presented between using the $k$-NN and $jk$-NN algorithms and showed that $jk$-NN enhanced accuracy when using the same verification system for both methods. The accuracy increased by 2.02% for the SVC2004 database, by 0.59% for the MCYT-100 database, by 0.73% for the SigComp’11 database, and by 10% for the SigComp’15 database.

The main parameters of $jk$-NN are $j$, $k$, reference count, and threshold. I showed that 15 references, a threshold of 1.5, would provide promising results based on the proposed evaluation phase’s experimental results. The case was different for the $j$ and $k$ values since they behaved differently for different circumstances. For $j$, the best results centered around the value of 5. For $k$, the optimal value requires careful consideration to achieve optimal results. Using these details, I have proposed an online $jk$-NN signature verification method that uses the preferred parameter values and calculates the optimal $k$ value for each signer. This method is realistic and generates real-life scenarios in which only the references of the signers are available. The accuracy of the approach proposed is encouraging. The achieved AER was 8% for SVC2004, 3.26% for MCYT-100, 13% for SigComp’15, and 2.22% for SigComp’11.

A combination of the $k$-nearest neighbor and dynamic time warping algorithms is presented in this thesis as an online signature verification system. The system is presented in the famous SVC2021 competition for both DeepSignDB and SVC2021_EvalDB databases. The proposed system achieved a very competitive results specially for mobile scenario online signatures where I achieved first place.
for the DeepSignDB and third for the SVC2021.EvalDB databases. These promising results also showed that the system could be improved to adopt more scenarios and achieve higher accuracy.
Chapter 6

Applications, Conclusion and Future work

The results of this dissertation are based on numerous experiments which were built during the last few years. The majority of these experiments were developed and tested using the SigStat project [Kovari et al., 2021]. The SigStat project is a notable contribution to this dissertation and for the researchers in the field. The first section of this chapter discusses the project in detail and elaborates my contribution to it. The second section of this chapter is devoted to summarizing the dissertation’s results and discussing future work.

6.1 Application of the results

This section presents the SigStat project, its structure, and my contributions to the project.

6.1.1 SigStat project introduction

SigStat is an open-source .NET class library for offline and online signature verification. The objective of SigStat is to provide a .NET class library for anyone interested in the field of signature verification. Together with my supervisor, we wanted to create a modular system, which breaks down the verification system
into separate steps. This approach allowed our students (and enables researchers) to focus on the improvement of subtasks instead of coping with the whole verification process. We also wanted to share not only our results but our algorithms and implementations. Most existing systems in the data processing area work with generic data types like vectors and multidimensional arrays [TFF, 2021][ML, 2021]. Within the SigStat system, we took advantage of the fact that we know that we are working with signatures. So we weren’t just trying to fit into some generic framework like machine learning models, but to provide a strongly typed framework for realizing, benchmarking and debugging signature verifiers. Our goals were the following:

- The system should support our research work. It should aid us in creating complex benchmarks for verification systems and try different combinations and algorithms.

- New students or researchers should be able to join and contribute smoothly.

- The system should be strongly typed, the main domain elements like signees and signatures should be represented directly in the system.

- The system should be open source and available for the community.

### 6.1.2 Project structure

The SigStat project grants developers a way to easily load several popular public databases by providing Loaders for them. The signatures in the database can be then processed using some transformation steps. These transformation steps may alter existing features of the signatures or generate new ones using different aggregation approaches. At the end of a pipeline, signatures may be sent to different classifier implementations, that can build signee specific profiles and make decisions about the origin of future samples. The project is divided into sub-folders that contains the relevant classes. Figures 6.1 and 6.2 show some examples.
6.1.3 My contribution to the project

The development of the SigStat project started almost parallel with my Ph.D. studies. I have added, modified, and tested several algorithms in the project related to
online signature verification during the last four years. It includes database loaders, preprocessing algorithms, helper classes, and classification algorithms. Loaders are used to acquire signatures from known databases, assign each signature to a signee and read their corresponding feature information. For this purpose I have created the following loaders:

- Loader for the SigWiComp’15 database ($\text{SigComp15GermanLoader}$).
- Loader for the sigComp’11 Chinese set ($\text{SigComp11ChineseLoader}$).
- Loader for the MCYT database ($\text{MCYTLoader}$).

Some benchmarks and algorithms were also implemented to test and evaluate signature verification systems. Up till now, I have implemented the following benchmarks and algorithms:

- $\text{OnlineRotationBenchmark}$ that tests all the rotation algorithms effects on the signature verification algorithms. It allows the user to test several rotation algorithms within any classifier to study its effect or to choose the optimal algorithm for his work.
- Rotation normalization algorithm ($\text{NormalizeRotation}$).
- Orthogonal rotation algorithm ($\text{OrthogonalRotation}$).
- Shifting algorithm ($\text{Shifting}$). Several techniques were used to translate the signature. It gives a variety of options for the user to transform the signature in the most suitable approach for the tested classifier.
- Algorithm for signature resampling using lower sampling frequency ($\text{Resampling}$). It allows the user to down-sample any signature and reduce the signature points number, which could be beneficial in many scenarios.
- I have implemented a $z$-normalization preprocessing method. The algorithm is very promising compared to the other scaling and shifting algorithms. It is implemented in a way that can be integrated with any classifiers as a preprocessing step.
Some classifiers were also created or extended by me to support my experiments and to present more classifiers for the research community, such as:

- Optimal DTW classifier that finds and uses the optimal values of the parameters used in the DTW classifier (OptimalDtwClassifier).

- Classifier that uses different DTW windows (DtwPyWindow). It allows the user to run the DTW algorithm using any window value, test the best value, or statically measure its effect.

- I added extra attributes to existing classes that can be used for specific benchmarks and algorithms.

- I created mathematical algorithms that can be used for statistical analysis and classification algorithms. These algorithms provide the developer with several statistics like signature points number and average, signer’s average signature points, signer’s optimal sampling frequency, different minimum and maximum statistics for signatures and signers features, similarity measurements related statistics, and more similar data, which are all made publicly available.

- I have created a $jk$-NN classifier and testing code for it. The classifier can be used with different values for each parameter, calculate the optimal value of $k$ for each signer, export each combination’s results to compare the results, and be integrated with other verifiers.

- I have implemented functions to test the effect of the sampling frequency on the verification system using different combinations. Several algorithms within this verifier can also be implemented to provide the researchers with numerous statistics and results, making it easier to understand and analyze the effect of the sampling frequency. Besides the verifier, it can also be used as a preprocessing step for any other verifier.

- I have implemented a signer-dependent sampling frequency-based classifier. It allows the user to choose the optimal samples selection approach for any database.
• I have implemented a classifier based on a combination of both $k$-NN and DTW algorithms.

• I have implemented a code for testing the optimal sampler.

• I have implemented a signer statistics helper. It is used to assign all the previous statistics and results to each signer, making it easier to use them without calculating everything again and separately.

### 6.2 Summary and future work

The main results of my dissertation are organized into three theses. The results are proven by experiments and compared to the literature. Most results were published or are under publication in respected international journals.

A systematic evaluation of the main algorithms in the field was presented in this work. A new approach using the effect of the sampling frequency of the input device and the number of signature points was also presented and analysed. A combination of $k$-NN and other algorithms was applied for the verification process. The previous results and combinations were used to build competitive online signature verification systems. Some of these systems were used to participate in well-reputed signature verification competitions and achieved good results. The summary of the theses results is presented in Appendix A.

Almost everything presented here is part of a still ongoing research. Some of the results are still under evaluation and development. My aim is to achieve more accurate and competitive results in the future. Many directions are possible for the future work, which are summarized as the following:

• Implementation of new signature verification methods using deep learning approach.

• Combine the best approaches into one strong verification system and analyse the results.

• Development of online-offline hybrid signature verification approach.
• Using more signature features that can be calculated and extracted from the original features.

• Apply the presented experiments using other features like altitude and azimuth.
Appendix A

Summary of the theses

In this appendix, I summarize the results of each thesis point and its related publication (see Table A.1).

Table A.1: Summary of related own publications

<table>
<thead>
<tr>
<th>Publication</th>
<th>Thesis I</th>
<th>Thesis II</th>
<th>Thesis III</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Saleem et al., 2021]</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saleem and Kovari, 2018b</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saleem and Kovari, 2018a</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saleem and Kovari, 2019b</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saleem and Kovari, 2019a</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saleem and Kovari, 2020d</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saleem and Kovari, 2020e</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Szucs et al., 2021</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saleem and Kovari, 2020b</td>
<td>●</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>Saleem and Kovari, 2020c</td>
<td>●</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>Saleem and Kovari, 2021e</td>
<td>●</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>Saleem and Kovari, 2021c</td>
<td>●</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>Saleem and Kovari, 2020a</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Saleem and Kovari, 2021d</td>
<td>●</td>
<td></td>
<td>●</td>
</tr>
<tr>
<td>Tolosana et al., 2021c</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saleem and Kovari, 2021b</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tolosana et al., 2021b</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saleem and Kovari, 2021a</td>
<td>●</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A.1 Summary of thesis I.

In this thesis, I investigated and analyzed the existing algorithms in the field, presented a systematic evaluation of preprocessing approaches in online signature verification, and built a verification system that combines the most significant approaches that minimized the error rate. I conducted extensive evaluations of some of the most popular design choices. A signature verifier was created for all linear combinations of the introduced methods, including some additional choices such as feature selection or the distance function used in conjunction with DTW. The resulting 42,336 verifier configurations were tested on five different publicly available databases. Supported by over 211,680 experiments, the results of this thesis are the following:

Sub-thesis I.1: Effect of feature selection resampling

In this sub-thesis, I investigated the effect of selecting the training set and the resampling on the accuracy of the verification system. To show the degree of bias, I introduced three additional new deterministic approaches for training set selection. The following four approaches have been used in our experiments to select the test set from an ordered list of genuine signatures of a given signee: first 10, last 10, odd 10, and even 10.

For resampling, my approach calculates a consistent time slot length for the signature based on the expected number of samples and the original length of the signature and estimates the new sample values based on the original data using an interpolation technique. In addition to the original sampling rate, four different sample counts (50, 100, 500, and 1000) were used to achieve both upsampling and downsampling. To evaluate the results, I compared the results of individual signature verifier configurations that differed only in their sample counts. The values of the other features are calculated at the new timestamps by interpolation. Two interpolation types were used in our experiments: cubic and linear. The results of the sub-thesis suggest that:
• Using the common method of selecting the first 10 genuine signatures as references is highly biased and should be avoided, especially when the signatures are sorted by their acquisition time.

• The most significant difference (8.85%) can be seen in the database SVC2004. Here, the best average ERR (18.07%) was achieved using the first signatures with odd indices in the training set, while the worst results (26.92%) were achieved by using the first ten signatures for training.

• Resampling signatures to a fixed number of samples with linear or cubic interpolation only improved the result in a negligible number of cases while all other cases had a negative impact; the EER increased in four out of five databases. In the case of the MCYT100 database, the error rate also increased in 99% of all the cases, compared to the verification results of the original sampling rates.

Sub-thesis I.2: Handling of pen-up strokes

In this sub-thesis, I investigated the effect of pen-up stroke handling methods. Both approaches of removing the pen-up strokes and filling in the missing data with interpolated points were tested. The values for the features at the new timestamps were calculated either by linear or by cubic spline interpolation. To assess the effects of this operation, I compared pairwise verification results of the non-filtered and filtered configurations.

I also compared three approaches for handling pen-up strokes, namely using the original pen-up strokes, using only pen-down strokes, and interpolation-generated pen-up strokes. The results of the sub-thesis suggest that:

• Removing pen-ups is always beneficial for the SigComp’11 Dutch and Chinese data sets, and it also has a positive effect in 78% of all the cases for the MCYT100 data-set.

• Existing pen-up strokes did not significantly contribute to improving the classification results in the examined databases. Filling in pen-up strokes with synthetic interpolated data did mostly negatively affect the results.
• Using only pen-down strokes may improve the verification accuracy in most cases.

**Sub-thesis I.3: Distance function within DTW**

In this sub-thesis, I evaluated the Manhattan and Euclidean distance measurement methods of Manhattan for $DTW_D$ similarity measurements in online signature verification. The effectiveness of the distance functions is compared in terms of two aspects. The first is the percentage of cases in which a distance function improves the equal error rate compared to the other. The second aspect is the quantity of this improvement. The results of the sub-thesis suggest that:

• The ratio of cases when the usage of Manhattan distance improved the equal error rate compared to Euclidean distance usage is higher than the reverse.

• The usage of Manhattan distance resulted in larger improvements.

**Sub-thesis I.4: Effects of translation and scaling**

In this sub-thesis, I investigated the translation and scaling methods and proposed a novel and a competitive combination of both preprocessing methods.

I introduce two approaches to scaling. The first scales the values into a fixed interval, while the other approach scales the values based on their standard deviation. In both cases, scaling is executed on the spot such that the minimum value of the scaled features is always fixed.

The two most commonly used preprocessing approaches that combine scaling and translation are $z$-normalization and scaling into the $[0,1]$ range (min–max normalization). These approaches are included in this study through a combination of the previously introduced translation and scaling methods. In addition, the distinction between the two allows me to introduce a new combination, where values are first scaled into the $[f_{\min}: f_{\min} + 1]$ range and then aligned to their center of gravity. The results of the sub-thesis suggest that:

• The best approaches take advantage of both types of translation and scaling algorithms.
Chapter A. Summary of the theses

• z-normalization results are superior to traditional min-max normalization.

• The proposed novel combination called centered min-max normalization, where values are first scaled into the \([f_{min}, f_{min} + 1]\) range and then aligned to their center of gravity yielded competitive results to that of z-normalization.

• Using these approaches in the preprocessing step highly affects the verification accuracy comparing to the other verification step algorithms.

Sub-thesis I.5: The effects of rotation normalization

In this sub-thesis, I investigated the rotation normalization effect on the verification accuracy. The rotation approach taken here is based on that of Xia et al. [Xia et al., 2017]. This method uses the \(X\) and \(Y\) feature vectors of a signature. The rotation methods were tested on five different databases. The results of the sub-thesis suggest that:

• Rotation normalization worsened the classification results in over 80% of the total cases and could not produce any performance improvement in three data-sets.

• The data suggests its usage to be more counterproductive in cutting-edge classifiers.

Sub-thesis I.6: Feature selection

In this sub-thesis, I investigated the effect of selecting the features on online signature verification systems. All possible combinations of the horizontal position, vertical position, and pressure features, namely \{X\}, \{Y\}, \{P\}, \{X, Y\}, \{X, P\}, \{Y, P\}, and \{X, Y, P\} have been evaluated.

Because of the differences between the databases, one cannot predict which individual feature has the best discriminative power; however, several single features can be grouped. The results of the sub-thesis suggest that:
DTW is beneficial in utilizing multiple features; therefore, the usage of all three features \((X, Y, P)\) together yields the highest accuracy compared to other feature sets in all the databases.

I recommend using \(X\), \(Y\), and \(P\) together to obtain the best results. Although the average EER of the remaining configurations may be lower for other feature combinations, the best EER was always achieved when the \(XYP\) feature combination was used.

Sub-thesis I.7: Systematic approach for online signature verification

In this sub-thesis, I took advantage of the previous observations of the effect of the main algorithms used for online signature verification to present a new competitive system.

According to previous observations, I can define several configurations that yield near-optimal results for most databases. Such a verifier should consider all three features \((X, Y, \text{and } P)\), use the Manhattan distance to calculate the \(DTW_D\) score of signature pairs, and remove the pen-up durations without filling these with artificial data. Resampling and rotation normalization should also be omitted. The best location and scale normalization techniques were \(z\)-normalization and the proposed centered min-max normalization. The results of this sub-thesis showed that:

- The proposed online signature verification system showed competitive results compared to the state as shown in Table A.2.

<table>
<thead>
<tr>
<th>Database</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCYT-100</td>
<td>1.35%</td>
</tr>
<tr>
<td>SVC2004</td>
<td>2.33%</td>
</tr>
<tr>
<td>SigComp’11 (Dutch)</td>
<td>0.72%</td>
</tr>
<tr>
<td>SigComp’11 (Chinese)</td>
<td>0.64%</td>
</tr>
<tr>
<td>SigWiComp’15</td>
<td>5.01%</td>
</tr>
</tbody>
</table>

I suggest using the introduced centered min-max normalization, which may be an alternative to \(z\)-normalization and yielded competitive results.
A.2 Summary of thesis II.

In thesis II, I investigated the direct effect of the sampling rate of the input signals on the accuracy of online signature verification systems without using interpolation techniques and proposed a signer-dependent online signature verification system.

The summary of the thesis results are the following:

Sub-thesis II.1: The effect of sampling rate on the verification accuracy

In this sub-thesis, I investigated the effect of the input device sampling rate and signature points number on the verification accuracy and presented an online signature verification using signature down-sampling.

I conducted thousands of measurements on five different public signature databases, and the results of the experiments showed a different behavior from the mentioned expectations. I did not use interpolation to avoid its effect on the results. The relation between the sampling frequency and the error rate was not monotonous in the majority of the cases; however, the error rate had a local minimum. Moreover, this local minimum was achieved in a similar range for several databases. Supported by 5600 experiments, the results of the sub-thesis suggest that:

- One can obtain better results by decreasing the sampling rate and the average number of sample points of the databases.
- The optimal sampling frequency should be between 15 Hz and 50 Hz.
- The optimal signature sample points for online signatures is between 60 and 240 points.
- Using frequencies lower than these ranges greatly decreased the accuracy, whereas using higher frequencies decreased or did not affect the accuracy in 92.5% of the configurations.
- Using sampling frequencies higher than 100 Hz will not improve the accuracy of the systems but will instead increase the computational cost of the verification.
Sub-thesis II.2: Accuracy improvement using signer-dependent sampling frequency

In this sub-thesis, I proposed a signer-dependent online signature verification system. The proposed technique is based on choosing the best sampling frequency for each signer before starting the verification process. It uses only the references for choosing the signer optimal sampling frequency, simulating a real-life scenario where no more genuine or forged signatures are available. The results of the sub-thesis suggest that:

- Using signer-dependent sampling frequency increases the system’s accuracy up to 8.4% comparing to the database common sampling frequency.
- The proposed system was tested using 500 different tests, the accuracy improved in around 80% of them.
- Using the optimal preprocessing method and sample number, the accuracy improved 92% of the tests.

Sub-thesis II.3: Signature verification system using signature down-sampling and signer-dependent sampling frequency approaches.

Table A.3: The error rate for each database.

<table>
<thead>
<tr>
<th>Database</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCYT-100</td>
<td>1.8%</td>
</tr>
<tr>
<td>SVC2004</td>
<td>1.6%</td>
</tr>
<tr>
<td>SigComp’11 (Dutch)</td>
<td>0.6%</td>
</tr>
<tr>
<td>SigComp’11 (Chinese)</td>
<td>0.1%</td>
</tr>
</tbody>
</table>

In this sub-thesis, I proposed an online signature verification system based on the previously concluded results in sub-theses 1 and 2. The proposed verification system achieved competitive accuracy results compared to the state of the art are summarized in Table A.3.
A.3 Summary of thesis III.

In thesis III, I investigated the $k$-nearest neighbor algorithm for online signature verification, improved it by using an optimized $jk$-NN algorithm, and created an online signature verification using the combination of $k$-NN and DTW algorithms. The summary of the thesis results are the following:

Sub-thesis III.1: Evaluation on the effect of the algorithm parameters

I have studied and evaluated the main parameters of the $jk$-NN algorithm using hundreds of experiments to eliminate the effect of irrelevant data. Furthermore, I built a $jk$-NN online signature verification system. The results of the sub-thesis suggest that:

- Better verification accuracy can be achieved using more than 10 original signatures as references, particularly between 13 and 15.
- A threshold $\theta=1.5$ provides the best results for the $jk$-NN algorithm for online signature verification.
- $jk$-NN enhanced accuracy compared to the $k$-NN algorithm when using the same verification system. The accuracy increased by 2.02% for the SVC2004 database, by 0.59% for the MCYT-100 database, by 0.73% for the SigComp’11 database, and by 10% for the SigWiComp’15 database.
- For $j$, the best results are centered around the value of 5. For $k$, the optimal value requires careful consideration to achieve optimal results.

Sub-thesis III.2: Optimized $jk$-NN algorithm for real-life scenarios signature verification

In this sub-thesis, I introduced a new optimized formula of the $jk$-NN algorithms, which is based on the minimum value of FRR reached within the training set to choose the best value of $k$, calculated in real-life circumstances where a certain number of signature references are available and used for this purpose. Using the previous evaluation of the algorithm parameters, the values $j = 5$ and $\theta = 1.5$ are
used with 15 reference signatures in the proposed algorithm. The results of the sub-thesis suggest that:

The proposed method is ideal for real-life scenarios in which only the references of the signers are available. The achieved accuracies of the method are encouraging considering the applied scenario and can be summarised in Table A.4

<table>
<thead>
<tr>
<th>Database</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCYT-100</td>
<td>3.26%</td>
</tr>
<tr>
<td>SVC2004</td>
<td>8%</td>
</tr>
<tr>
<td>SigComp’11</td>
<td>2.22%</td>
</tr>
<tr>
<td>SigWiComp’15</td>
<td>13%</td>
</tr>
</tbody>
</table>

**Sub-thesis III.3: $k$-NN and DTW for mobile scenario online signature verification**

In this sub-thesis, I presented an online signature verification system based on a novel combination of the DTW and the $k$-NN algorithms. The system considers local thresholds to classify signatures and uses DTW and $k$-NN algorithms in the classification phase. DTW is used as a distance measurement between the signatures. The $k$-NN algorithm is used to select the reference signatures and calculate the upper and lower threshold, which plays a significant part in calculating the prediction of the tested signature. The results of the sub-thesis suggest that:

The proposed system was presented in the SVC2021 signature verification competition for both DeepSignDB and SVC2021_EvalDB databases and achieved the following:

- In the development stage, the proposed system ranked 2nd overall and achieved 7.74% EER in Task1, 5.81% in Task 2 (1st place), and 7.71% in Task 3 (3rd place).

- In the evaluation stage, the proposed system achieved 13.29% in Task2 (bronze medal) and 4th overall.
Bibliography


BIBLIOGRAPHY


