Continuous Authentication using Stylometry

by

Marcelo Luiz Brocardo

B.Sc. of Computer Science, Regional University of Blumenau, Brazil, 1995

M.Sc. of Computer Science, Federal University of Santa Catarina, Brazil, 2001

A Dissertation Submitted in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

Continuous Authentication using Stylometry

by

Marcelo Luiz Brocardo

B.Sc. of Computer Science, Regional University of Blumenau, Brazil, 1995

M.Sc. of Computer Science, Federal University of Santa Catarina, Brazil, 2001

Supervisory Committee

Dr. Issa Traoré, Supervisor

(Department of Electrical and Computer Engineering, University of Victoria)

Dr. Lin Cai, Departmental Member

(Department of Electrical and Computer Engineering, University of Victoria)

Dr. Venkatesh Srinivasan, Outside Member

(Department of Compute Science, University of Victoria)

Supervisory Committee

Dr. Issa Traoré, Supervisor

(Department of Electrical and Computer Engineering, University of Victoria)

Dr. Lin Cai, Departmental Member

(Department of Electrical and Computer Engineering, University of Victoria)

Dr. Venkatesh Srinivasan, Outside Member

(Department of Compute Science, University of Victoria)

ABSTRACT

Static authentication, where user identity is checked once at login time, can be circumvented no matter how strong the authentication mechanism is. Through attacks such as man-in-the-middle and its variants, an authenticated session can be hijacked later after the initial login process has been completed. In the last decade, continuous authentication (CA) using biometrics has emerged as a possible remedy against session hijacking. CA consists of testing the authenticity of the user repeatedly throughout the authenticated session as data becomes available. CA is expected to be carried out unobtrusively, due to its repetitive nature, which means that the authentication information must be collectible without any active involvement of the user and without using any special purpose hardware devices (e.g. biometric readers). Stylometry analysis, which consists of checking whether a target document was written or not by a specific individual, could potentially be used for CA. Although stylometric techniques can achieve high accuracy rates for long documents, it is still challenging to identify an author for short documents, in particular when dealing with large author populations.

In this dissertation, we propose a new framework for continuous authentication using authorship verification based on the writing style. Authorship verification can be checked using stylometric techniques through the analysis of linguistic styles and writing characteristics of the authors. Different from traditional authorship verification that focuses on long texts, we tackle the use of short messages. Shorter authentication delay (i.e. smaller data sample) is essential to reduce the window size of the re-authentication period in CA. We validate our method using different block sizes, including 140, 280, and 500 characters, and investigate shallow and deep learning architectures for machine learning classification. Experimental evaluation of the proposed authorship verification approach based on the Enron emails dataset with 76 authors yields an Equal Error Rate (EER) of 8.21% and Twitter dataset with 100 authors yields an EER of 10.08%. The evaluation of the approach using relatively smaller forgery samples with 10 authors yields an EER of 5.48%.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Dr. Issa Traoré, for his enlightening guidance, support, and inspiring advice throughout the course of this work.

Thank you to Isaac Woungang and to my committee members, Dr. Lin Cai and Dr. Venkatesh Srinivasan, for all their valuable advice and critical feedback. My gratitude is extended to the external examiner Dr. Fatos Xhafa for putting time and effort in the evaluation of my work.

I extend my sincere thanks to all those who participated as a volunteer in the forgery experiment, thank you.

To my classmates Sherif Saad, Bassam Sayed, Abdulaziz Aldribi and Asem Kittaneh, thank you for the cooperative and friendly environment that undoubtedly played an important rule during all my PhD program.

Thanks to my friends, Paul Mohapel and Joana Gil Mohapel, for the support and help during my stay in Canada.

Also, I would like to acknowledge the WestGrid and Compute/Calcul Canada by providing computing resources, and also the financial support received from the Natural Sciences and Engineering Research Council of Canada (NSERC) through a Vanier scholarship and also from the National Council for Scientific and Technological Development (CNPq - Brazil).

I would especially like to express my gratitude to my wife, who has always supported me and helped me overcome the difficulties and to my children, Wellington and Giulia, for understanding my absence in their lives when my work prevents us from sharing important moments of life. Without the support of all the members of my family, I would have never finished this thesis.

Marcelo Luiz Brocardo, Victoria, BC, Canada

DEDICATION

I dedicate this work to my mother (in memory) that always encouraged me to study.

Chapter 1

Introduction

The way we handle information has dramatically changed over the past few years. Exchange of information between organizations is expanding and growing rapidly, not only among computers but also between cell phones and tablets. The use of electronic documents has several advantages over the use of paper documents, such as ease of administration, copying, storage and transmission. Electronic information is accepted and treated naturally in various business relationships between companies, citizens and governments. These technological advances have restructured the economic model, from an industrial model to an information model. However, the vulnerability in the access and storage of electronic information together with the risks associated with their misuse, have motivated administrators to seek mechanisms to counteract this fragility. Protecting electronic information against unauthorized access has become a critical issue.

1.1 Context

Authentication mechanisms represent the lock to modern computer networks with password-based authentication being the most widely used mechanisms. However, several high-profile hacking incidents which occurred recently have reminded us that initial authentication at login time can be circumvented no matter how strong the authentication mechanism is. Through attacks such as man-in-the-middle and its variants, an authenticated session can be hijacked later after the initial login process has been completed. In the last decade, continuous authentication (CA) using biometrics has emerged as a possible remedy against session hijacking. CA consists of testing the authenticity of the user repeatedly throughout the authenticated session as data becomes available. Continuous authentication is expected to be carried out unobtrusively, due to its repetitive nature, which means that the authentication information must be collectible without any active involvement of the user and without using any special purpose hardware devices (e.g. biometric readers).

Emerging behavioural or cognitive factors such as mouse dynamics, keystroke dynamics, and stylometry are good candidates for CA because data can be collected passively using standard computing devices (e.g. mouse and keyboard) throughout a session without any knowledge of the user. One of the main issues with these technologies is that their accuracy tends to degrade significantly as the amount of data involved in the authentication decreases. However, shorter authentication delay (i.e. smaller data sample) is essential to reduce the window of vulnerability of the system. Therefore, there is a need for the above modalities to develop new analytical models that will achieve high accuracy while maintaining acceptable authentication delays.

Based on the above considerations, it is of paramount importance to develop a new authentication methodology that can be non intrusive, efficient, and transparent. We believe that developing continuous authentication approach based on authorship analysis will contribute to achieving this goal. Specifically, our goal in this research is to develop a new stylometric model for continuous authentication. While forensics authorship identification using stylometry has been widely studied, authentication using that modality is still in its infancy.

1.2 Problem Statement and Research Objectives

The writing style is an unconscious habit, which varies from one author to another in the way he/she uses words and grammar to express an idea. The patterns of vocabulary and grammar could be a reliable indicator of the authorship. The linguistic characteristics used to identify the author of a text is referred to as stylometry [44,76]. Although the writing style may change a bit with time [22], each author has a unique stylistic tendency.

Forensic authorship analysis consists of inferring the authorship of a document by extracting and analyzing the writing styles or stylometric features from the document content. Authorship analysis of physical and electronic documents has generated a significant amount of interest over the years and led to a rich body of research literature [2, 23, 66, 90]. Authorship analysis can be carried out from three different perspectives, including, authorship attribution or identification, authorship verification, and authorship profiling or characterization. Authorship attribution consists of determining the most likely author of a target document among a list of known individuals. Authorship verification consists of checking whether a target document was written or not by a specific individual. Authorship profiling or characterization consists of determining the characteristics (e.g. gender, age, and race) of the author of an anonymous document.

Among the above three forms of stylometry analysis, authorship verification is the most relevant to CA, as user identity verification is central to any authentication system. However, according to Koppel et al., "using stylometry verification is significantly more difficult than basic attribution and virtually no work has been done on it, outside the framework of plagiarism detection" [66]. Most previous works on authorship verification focus on general text documents. However, authorship verification for online documents can play a critical role in various criminal cases such as blackmailing and terrorist activities, to name a few.

Similar to forensic authorship verification, authentication consists of comparing sample writing of an individual against the model or profile associated with the identity claimed by that individual at login time (i.e. 1-to-1 identity matching). While a rich body of literature has been produced on authorship attribution/identification and authorship characterization using stylometry, limited attention has been paid to authorship verification [2, 23, 66, 90].

In particular, stylometry-based authorship verification for online documents (e.g. emails, tweets) pose significant challenges because of the unstructured nature of such documents. Furthermore, a key requirement of CA is that (repeated) authentication decisions should occur over short time period or short text or messages. Stylometry analysis of short messages is challenging because of the limited amount of information available for decision making. Likewise, most of the stylometry analysis approaches proposed in the literature use relatively large document size which is unacceptable for continuous authentication.

Another important challenge to address when using stylometry for CA is the threat of forgery. An adversary having access to writing samples of a user may be able to effectively reproduce many of the existing stylometric features. It is essential to integrate specific mechanisms in the authentication system that would mitigate forgery attacks.

The goal of the proposed research is to develop a new framework for continuous authentication using stylometry. This will require developing a robust authorship verification model for short online documents, since verification is the central factor in any authentication system.

The proposed research dissertation is articulated around four main tasks as follows:

1. To analyze the text and obtain identical structural data, and to extract patterns of authorial attributes in order to address the problem of authorship verification;

2. To propose a supervised learning technique combined with a stylometric analysis approach to check the identity of the author of a short online document;

3. To investigate and propose an authorship verification method that achieves high accuracy classification;

4. To integrate authorship verification in a continuous authentication framework and test the proposed method against forgery.

1.3 General Approach

Our approach to address the above challenges it to explore new stylometric features and robust classifiers. In a general overview of the proposed approach, an online document is decomposed into consecutive blocks of short texts over which (continuous) authentication decisions happen. For each block of text, a feature vector is extracted based on all features. The classification model consists of a collection of profiles generated separately for individual users. The proposed system operates in two modes: enrolment and verification. Based on sample training data, the enrolment process computes the behavioral profile of the user using machine learning classification. For classification, this research investigates shallow and deep classifiers.

Shallow-structured architectures of machine learning have been widely used for authorship analysis of electronic documents [2, 23, 26, 56, 66, 68, 72, 101]. A shallow architecture refers to a classifier with only one or two layers responsible for classifying the features into a problem-specific class. Some examples of shallow classifiers with one layer include k-Nearest Neighbor (k-NN), Naïve Bayes, Hidden Markov Model (HMM), Principal Component Analysis (PCA), Logistic Regression (LR), and Support Vector Machines (SVM). Examples of shallow classifiers with two layers include SVM-Logistic Regression (SVM-LR), where the output of the SVM is submitted to a logistic function. It has been shown that shallow architectures can be effective in solving many stylometric analysis problems [68, 94].

Deep models, such as Deep Belief Network (DBN), have emerged as an alternative to shallow machine learning techniques [46]. Deep models try to imitate the brain using hidden layers with many neurons, and have been shown to be powerful analysis techniques in handwriting recognition, visual detection of objects, and speech recognition [16, 34, 48, 88]. DBN is a probabilistic generative model composed of many layers of non-linear processing stages and a softmax layer implemented at the final layer of the network used for classification. The Softmax layer in this case is composed of a shallow classifier, specifically a logistic function, which is a commonly used activation function for neural networks. The non-linear layers extract structures and regularities of the input features through an unsupervised learning method, and each layer's outputs are fed to the inputs of the next higher layer.

In this dissertation, we introduce new stylometry features families based on n-gram analysis and features merging process, and investigate SVM, LR, and SVM-LR, as candidate shallow classifiers. In addition, we present a stylometry-based authorship verification model based on the Gaussian-Bernoulli Deep Belief Network, which uses Gaussian units in the visible layer to model real-valued data [45, 71]. To our knowledge, this is the first time that DBN is used for stylometry-based authorship analysis.

The proposed approach is evaluated experimentally by computing the following performance metrics:

- False Acceptance Rate (FAR): measures the likelihood that the system may falsely recognize someone as the genuine person;
- False Rejection Rate (FRR): measures the likelihood that the system will fail to recognize the genuine person;
- Equal Error Rate (EER): corresponds to the operating point where FAR and FRR have the same value.

Experimental evaluation is conducted using the Enron emails dataset and a micro-messages dataset based on Twitter feeds. Furthermore, a forgery dataset was created as part of this research by collecting simulated attacks against 10 users' profiles. Different block sizes were tested including 140, 280, and 500 characters on the datasets mentioned above. The evaluation yielded EER ranging from 8.21% and 10.08% for block sizes of 500 and 280 characters, which is very encouraging considering the existing works on authorship verification using stylometry.

1.4 Research Contributions

The contributions of this research can be described in the following points:

A new model for CA based on stylometry: The existing works on stylometry have focused primarily on identification and characterization. The first contribution of this research is to help bridge the gap in this area, by proposing an effective stylometric authorship verification approach that can be used for continuous authentication. The proposed model yields very encouraging results in addressing the main challenges faced by a continuous authentication system, which consist of the needs for short authentication delay, high authentication accuracy, and resilience to forgery. The performance achieved by the proposed model outperforms existing authorship verification approaches proposed in the literature.

A paper published in the proceedings of the 28th IEEE International Conference on Advanced Information Networking and Applications (AINA-2014) presents our framework for continuous authentication using stylometry. Further results were published in the Twelfth Annual International Conference on Privacy, Security and Trust (PST 2014) and in the Journal of Computer and System Sciences - Elsevier (JCSS).

New Feature Families: The second contribution of this research is to derive new stylometric features using new n-gram and feature merging models.

***N*-gram** is a type of lexical features that has proven to be efficient in capturing writing style. N-gram is a token formed by a contiguous sequence of characters or words. The proposed n-gram model analyzes n-grams and their relationship with the training dataset. A basic version of the n-gram model was published in the IEEE Intl. Conference on Computer, Information and Telecommunication Systems (CITS 2013) and received the best paper award [18]. An extended version of the same model was published later in the Journal of Networks (JNW).

Feature Merging consists of computing new features by merging existing ones. The proposed method merges a pair of features into a single feature that considers the information gain as selection criteria.

Datasets: The quantity of messages written by the same author in the available datasets is very small and insufficient to run the proposed stylometry experiments, which need at least 28,000 characters per author. As part of this work, a dataset was created by crawling messages of authors from Twitter. The Twitter dataset contains 100 English users and on average 3,194 twitter messages with 301,100 characters per author. Moreover, in order to assess the robustness of our proposed approach against forgeries attempts, a novel forgery dataset was collected as part of this research. Both datasets have been made available publicly for the research community.

Deep models: Deep models have been shown to be powerful analysis techniques in handwriting recognition, visual detection of objects, and speech recognition, exhibiting an effective encoding learning of a complex distribution in an unsupervised manner. The fourth main contribution of this thesis is to apply for the first time deep machine learning technique for the classification of the stylometry profiles.

1.4.1 List of papers

This section enumerates the complete list of papers published as a result of this work.

Journals:

1. Brocardo, Marcelo Luiz; Traore, Issa; Woungang, Isaac. **Authorship Verification of E-mail and Tweet Messages Applied for Continuous Authentication**. Journal of Computer and System Sciences, Elsevier, Available online 29 December 2014, ISSN 0022-0000.

2. Brocardo, Marcelo Luiz; Traore, Issa; Saad, Sherif; Woungang, Isaac. **Verifying Online User Identity using Stylometric Analysis for Short Messages**. Journal of Networks 9, no. 12 (2014): 3347-3355.

Conferences:

1. Brocardo, Marcelo Luiz; Traore, Issa. **Continuous Authentication using Micro-Messages**. Twelfth Annual International Conference on Privacy, Security and Trust (PST 2014), Toronto, Canada, July 23-24, 2014.

2. Brocardo, Marcelo Luiz; Traore, Issa; Woungang, Isaac. **Toward a Framework for Continuous Authentication using Stylometry**. The 28th IEEE International Conference on Advanced Information Networking and Applications (AINA-2014), Victoria, Canada, May 13, 2014.

3. Brocardo, Marcelo Luiz; Traore, Issa; Saad, Sherif; Woungang, Isaac. **Authorship Verification for Short Messages Using Stylometry**. Proc. of the IEEE Intl. Conference on Computer, Information and Telecommunication Systems (CITS 2013), Piraeus-Athens, Greece, May 7-8, 2013 (Best paper award).

1.5 Dissertation Organization

The remaining chapters of this dissertation are structured as follows.

Chapter 2 gives an overview of the literature underlying this research. It provides a quick introduction to continuous authentication and presents a generic architecture of a biometric system. Also, Chapter 2 introduces the stylometric authorship analysis and related works.

Chapter 3 describes our experimental evaluation method and settings, including the dataset used in the experiments, data preprocessing, and experimental procedure. Also, this chapter provides an explanation of the performance calculation method used in this research.

Chapter 4 discusses the most common writing characteristics used to create a profile that represents the style of an author. Furthermore, chapter 4 introduces Information Gain and Mutual Information as feature selection technique to reduce large feature space and eliminate redundant features.

Chapter 5 presents our proposed approach for continuous authentication using shallow classifiers. These classifiers include SVM, SVM-LR and LR.

Chapter 6 introduces a new method to merge a pair of random features into a single feature. Shallow classifiers are used to perform the experimental evaluation.

Chapter 7 investigates stylometry-based authorship verification using deep classifiers, specifically Deep Belief Network. In addition, this chapter assesses the strength of the proposed approach against forgeries.

Chapter 8 discusses the performance results obtained for all the experiments conducted using shallow and deep classifiers.

Chapter 9 concludes the dissertation by discussing the overall contribution of the research in the context of related work in the area. In addition, it outlines a number of ideas for future work.

Chapter 2

Background and Literature Review

In this chapter, we introduce authentication systems and continuous authentication. We also describe the state-of-the-art techniques in authorship analysis using stylometry. The authorship analysis using stylometry can be studied from three different perspectives, i.e, authorship attributions or identification, authorship characterization, and authorship verification.

This chapter is organized as follows. Section 2.1 discusses authentication systems, introduces biometric authentication, and sketches a generic architecture of a continuous authentication. Section 2.2 summarizes and discusses stylometry analysis related works. We summarize the chapter in Section 2.3.

2.1 Background on Authentication Systems

2.1.1 User Authentication

User authentication allows the verification of a user identity prior to granting him access to sensitive applications or resources. The user authentication mechanisms can be based on knowledge (something the user knows), possession (something the user has), or inherent factors (something the user is). Each authentication method defines the requirements for identities to be verified.

Authentication based on knowledge is the most widely used method to check the user identity and can be based on a simple password or a challenge/response system. Authentication mechanisms based on passwords are simple and inexpensive. However, some users tend to choose easy passwords, which can be easily guessed through dictionary or social engineering attacks. In addition, it is common that a user will interact with several systems and each one will require a password, leading the user to re-use the same password for multiple systems. Authentication based on a challenge/response system consists of prompting the user with a random set of questions, such as birth date, pet's name, and favorite places. During the login process, a random question is asked, and access is granted only if the answer is correct. While the cost to implement such authentication system is low, it is also highly vulnerable to attacks and could easily be broken.

Authentication methods based on possession depend on a physical object that the user has, for instance a smart card or a token. One-Time Password (OTP) tokens prevent an attacker from capturing and replaying the password because the system will require a different password for each session. The disadvantages include the cost of the physical objects and the possibility of these being lost.

The third type of authentication method consists of using characteristics that are intrinsic to the user, which are typically based on biometrics. Biometric systems are discussed in more details in the next section.

2.1.2 Biometric Authentication

Biometrics technologies are considered the most effective and accurate authentication system [5]. Biometric technologies are broadly categorized into physiological or behavioral biometrics depending on the type of unique characteristics (behavioral or physiological) that make up the biometrics. Physiological biometrics measure biological attributes and include fingerprint scan, face scan, iris scan, retina scan, etc. Behavioral biometrics measure habits and include signature scan, voice scan, keystroke dynamics, etc. Another behavioral feature that can be extracted from a person is the linguistic style employed during writing, which is referred to as stylometry. Physiological characteristics are static and could change only in extreme circumstances such as accidents or trauma. On the other hand, behavioral characteristics are evolving since numerous factors such as stress, health issues, or danger situations can potentially influence behavior and create imprecision in the system. The concern of most biometric authentication methods is the high cost of the hardware devices that are required to collect and analyze the data. Most (not all) behavioral biometrics require less expensive hardware devices than physiological biometrics.

We highlight the main physiological biometrics used in authentication systems:

- **Fingerprint** is one of the oldest and most widespread biometrics technologies [41]. The identification of a person by his fingerprint is done through the analysis of loops and arches from the finger, captured using a fingerprint scanner. Fingerprint biometric is used mainly for static user authentication.

- **Face** recognition is a process that identifies a person from a video source or thermal images [37]. The system extracts some facial features such as shape, pattern and positioning of the face in order to build a facial database. Recognition of faces from an uncontrolled environment is complex, as lightning conditions may vary immensely. Furthermore, facial expressions also vary from time to time, and a face can appear at different orientations and even be partially occluded at times. Also, people do change over time; wrinkles, beard, glasses and position of the head can affect the performance considerably.

- **Retina** biometric uses the vascular pattern of the human eye's to authenticate a person [75]. Although retina biometric produces one of the best results for authentication, the reader uses a sophisticated infra-red light to scan the eye, , which can be cumbersome. Highly secured facilities use retina biometric as a static user authentication method.

- **Iris** biometric extracts visible features from the pigmented ring around the eye's pupil [75]. Iris biometric requires a high-precision camera and is used for static authentication.

- **Hand Geometry** biometric uses the shape and length of fingers and knuckles as a measure [41]. A hand is placed in a specific position, typically guided, and a reader captures all measurements. It has been used for access control.

Behavioral biometrics are relatively recent compared with physiological biometrics and the most commonly used in user authentication are the following:

- **Gait** biometric measures the way a person walks and can vary from time to time due to changes, such as major shift in the body weight, or major injuries. Gait biometric features can be extracted by analysing a video or by collecting information from a floor sensor [39].

- **Keystroke** dynamic biometric is a behavioral biometric based on the analysis of the typing habits. The features include the typing rhythm extracted by measuring the dwell time (the time a keyboard key is pressed down) for a specific key and the fly time between keys [3].

- **Mouse dynamics** biometric captures the mouse movement characteristics and does not require a special hardware device for data collection. The features include information such as Movement Speed, Movement Direction, Action Type, Traveled Distance, and Elapsed Time [4].

- **Voice** biometric is a characteristic of a person and could be used more for verification than identification, because it is not unique to identify a single person from a large database [57]. The voice of a person may change if (s)he is sick, in a dangerous situation, or afraid. In addition, the voice may change significantly over the years, specially during puberty. One problem that could degrade this biometric system is the use of poor microphone to capture the voice.

- **Signature** biometric is related to the way a person signs her name. Paper-based signature is already widely accepted in many legal transactions. The feature set includes spatial coordinates, pressure, inclination, pen up/down and azimuth [51].

- **Stylometry** consists of the analysis of linguistic styles and writing characteristics of a person. The patterns of vocabulary and grammar could be a reliable indicator of the user identity. Detailed review on previous work on stylometry is presented in the Section 2.2.

Table 2.1 and 2.2 shows a comparison of different physiological and behavioral biometric systems, adapted from Jain et al. [57]. The following criteria could be used to select the best biometric solution to be applied for authentication or identification.

- **Universality:** indicates whether every person possesses the biometric characteristic;

- **Uniqueness:** indicates how unique and different the biometric characteristics are for each user among groups of users;

- **Permanence:** measures the effect on the system when the biometric characteristic changes over the years;
- **Measurability** (or collectability): expresses how difficult or time consuming it is to measure the biometric characteristic;
- **Acceptability:** measures how well a user accepts the technology;
- **Performance:** is measured in terms of speed, accuracy, and robustness;
- **Circumvention:** measures how easy it is to imitate or forge the biometric characteristics.

Table 2.1: Comparison of physiological biometric systems

Characteristics	Face	Fingerprint	Iris	Retina	Hand Geometry
Universality	High	Medium	High	High	Medium
Uniqueness	Low	High	High	High	Medium
Permanence	Medium	High	High	Medium	Medium
Measurability	High	Medium	Medium	Low	High
Performance	Low	High	High	High	Medium
Acceptability	High	Medium	Low	Low	Medium
Circumvention	Medium	Medium	Low	Low	Medium

Table 2.2: Comparison of behavioral biometric systems

Characteristics	Gait	Keystroke	Mouse Dynamics	Voice	Signature
Universality	Medium	Low	Low	Medium	Low
Uniqueness	Low	Low	Low	Low	Low
Permanence	Low	Low	Low	Low	Low
Measurability	High	Medium	Medium	Medium	High
Performance	Low	Low	Low	Low	Low
Acceptability	High	Medium	Medium	High	High
Circumvention	Medium	Medium	Medium	Low	Low

A biometric process typically involves three steps: enrolment, matching and decision (see Figure 2.1, adapted from [5]). During the enrolment phase, biometric sample is acquired by a sensing device from an individual, specific features are then extracted from the biometric sample and used to create a template/profile based on a mathematical representation of the raw biometric data. In the matching phase the new captured biometric data is compared against the user's template. A biometric system can be used both for identification and verification purposes. In an identification process, the system recognizes an individual by searching the templates of all the users in the database for a match through a one-to-many comparison. In contrast, a verification process validates the identity of a person by comparing the captured biometric data with the person's template through a one-to-one matching.

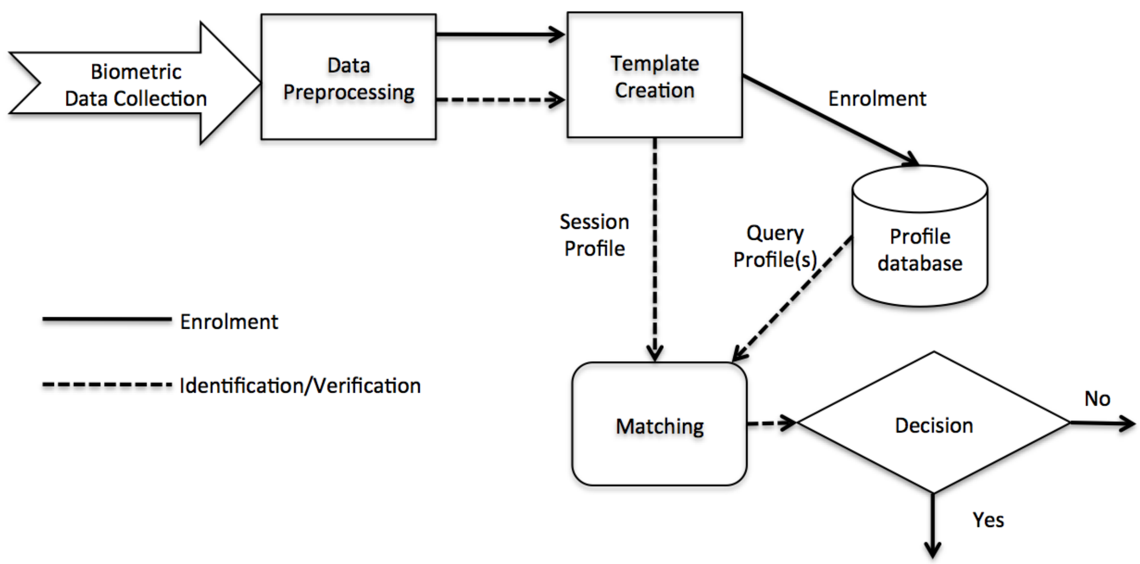


Figure 2.1: Generic architecture of biometric system

The similarity between an input X_i and the database template X_j is represented by the matching score or biometric score $S(X_i, X_j)$. The decision is made by comparing the matching score with a threshold t. If the score is higher than or equal to t, it is inferred that the sample belongs to the same person. Otherwise, it is inferred that the sample belongs to a different person. The threshold t can be tuned to minimize or maximize the acceptance or rejection of a person. Figure 2.1 shows the impact of choosing a different value for the threshold.

The following key metrics are traditionally used to evaluate the performance of biometrics systems:

- False Rejection Rate (FRR): measures the likelihood that the system will fail to recognize the genuine person. This metric is also referred as “Type I error”, False Non-Match Rate (FNMR), or False Positive Rate (FPR);

- False Acceptance Rate (FAR): measures the likelihood that the system may falsely recognize someone as the genuine person. This metric is also referred as “Type II error”, False Match Rate (FMR) or False Negative Rate (FNR);

- Equal Error Rate (EER): corresponds to the operating point where FAR and FRR have the same value.

Figure 2.2 illustrates how a threshold can affect FRR and FAR. When FAR is very high, the system is very susceptible to intrusions. On the other hand, high FRR indicates that the system rejects genuine users in high number. The problem is that FRR and FAR are inversely proportional, the reduction in one creates an increase in the other. So a trade-off must be made to identify the optimum operating point.

2.1.3 Continuous Authentication

Traditional approaches for user authentication consists of statically checking the user identity once, typically at login time. However, this may allow a hacker to hijack a session. Implementing a continuous authentication process, which consists of repeatedly verifying user identity during a session, has been advocated as a way to address the above mentioned limitation. The principle of continuous authentication is to monitor the user behavior during the session, while discriminating between normal and suspicious user behavior. In case of suspicious behavior, the user session

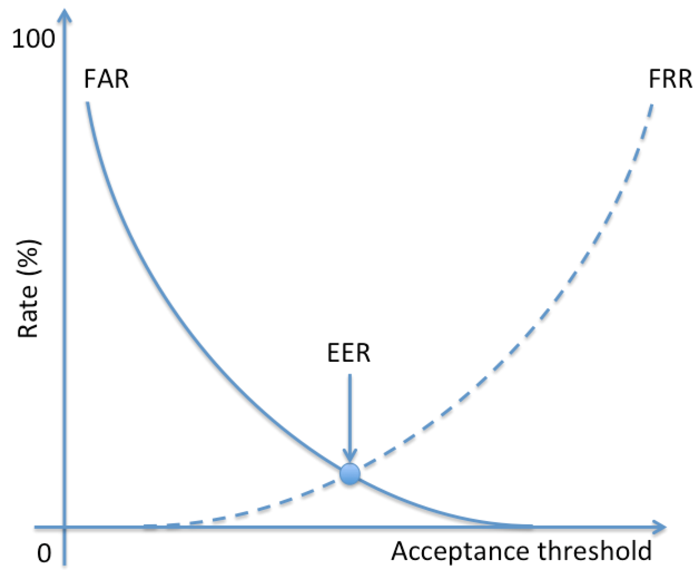


Figure 2.2: Relationship between FRR and FAR. This diagram demonstrates how a threshold can affect FRR and FAR. EER can be obtained by adjusting the classifier acceptance threshold, where FAR and FRR have the same value.

is closed, or an alert is generated. As shown in Figure 2.3 (adapted from [36]), the flag to prompt another authentication is based on time or the amount of data (delay between consecutive re-authentication). Continuous authentication has been applied for intrusion detection, network forensics, insider detection and session security [99]. CA involves several challenges including the need for low authentication delay, high accuracy, and the ability to withstand forgery.

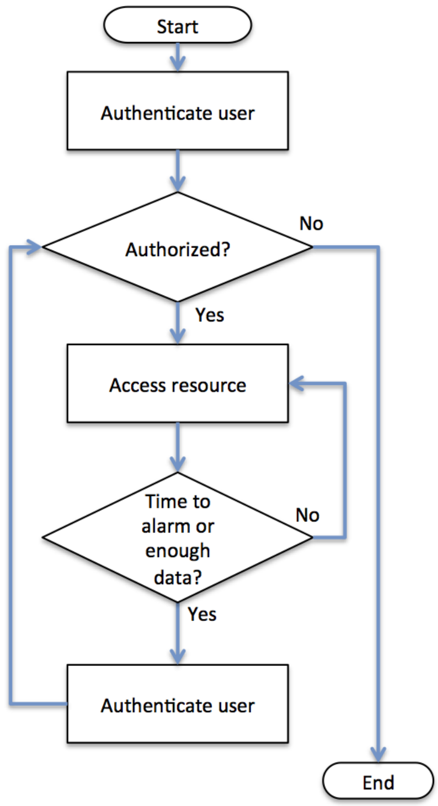


Figure 2.3: Generic architecture of continuous authentication system

2.2 Related Work on Stylometry Analysis

2.2.1 Overview

Authorship analysis using stylometry has so far been studied primarily for the purpose of forensic analysis. Writing style is an unconscious habit and the patterns of vocabulary and grammar could be a reliable indicator of the authorship. Stylometry studies typically target three different problems, including, authorship attribution or identification, authorship verification, and authorship profiling or characterization. Authorship attribution consists of determining the most likely author of a target document among a list of known individuals. Earliest successes in attempting to quantify the writing style were the resolution of disputed authorship of Shakespeare's plays by Mendenhall [78] in 1887 and the Federalist Papers by Mosteller and Wallace in 1964 [80]. Recently studies on authorship identification investigated ways to identify patterns of terrorist communications [1], the author of a particular e-mail for computer forensic purposes [54–56], as well as how to collect digital evidence for investigations [25] or solve a disputed literary, historical [80], or musical authorship [9, 19, 107]. Work on authorship characterization has targeted primarily gender attribution [27, 28, 87] and the classification of the author education level [59]. Authorship verification consists of checking whether a target document was written or not by a specific author. There are few papers on authorship verification outside the framework of plagiarism detection [66], and most of them focus on general text documents. In addition, the performance of authorship verification for online documents is affected by the text size, the number of candidates authors, the training set size, and the fact that these documents are in general quite poorly structured or written (as opposed to literary works). In subsequent subsections, we present related works on stylometry for authorship attribution, characterization, and verification.

2.2.2 Authorship Attribution or Identification

Authorship attribution follows typical biometric identification process, where the system recognizes an author through one-to-many comparison. The process consists of extracting features from sample texts and labeling the classes according to the authors of the documents. Typical features categories include lexical, semantic, syntactic and application specific. Authorship attribution is similar to text classification. A key difference, however, is that authorship attribution is topic-independent, while in text classification, the class labels are based on the topic of the document and the features include topic-dependent words.

Despite significant progress achieved on the identification of an author within a small group of individuals, it is still challenging to identify an author when the number of candidates increases or when the sample text is short as in the case of e-mails or online messages. For instance, while Chaski [25] reported 95.70% accuracy in their work on authorship identification, the evaluation sample consisted of only 10 authors.

Similarly, Iqbal et al. [53] achieved, when using k-means for author identification, classification accuracy of 90% with only 3 authors; the rate decreased to 80% when the number of authors increased to 10. Iqbal et al. [55] also proposed another approach named AuthorMiner, which consists of an algorithm that captures frequent lexical, syntactic, structural and content-specific patterns. The experimental evaluation used a subset of the Enron dataset, varying from 6 to 10 authors, with 10 to 20 text samples per author. The authorship identification accuracy decreased from 80.5% to 77% when the authors population size increased from 6 to 10.

Hadjidj et al. [42] used the C4.5 and SVM classifiers to determine authorship, and evaluated the proposed approach using a subset of three authors from the Enron dataset. They obtained as correct classification rates 77% and 71% for sender identification, 73% and 69% for sender-recipient identification, and 83% and 83% for sender-cluster identification, for C4.5 and SVM, respectively.

2.2.3 Authorship Characterization

Works on authorship characterization have targeted the determination of various traits or characteristics of an author such as gender, age, or education level. Authorship characterization is addressed as a text classification problem. The general approach consists of creating socio-linguistic clusters from documents written by the same population, and then inferring the group of an anonymous document.

Cheng et al. [27] investigated the author gender identification from text by using Adaboost and SVM classifiers to analyze 29 lexical character-based features, 101 lexical word-based features, 10 syntactic, 13 structural, and 392 functional words.

Evaluation of the proposed approach involving 108 authors from the Enron dataset yielded classification accuracies of 73% and 82.23%, for Adaboost and SVM, respectively.

Abbasi and Chen [1] analyzed the individual characteristics of participants in an extremist group web forum using decision tree and SVM classifiers. Experimental evaluation yielded 90.1% and 97% success rates in identifying the correct author among 5 possible individuals for decision tree and SVM, respectively.

Kucukyilmaz et al. [73] used k-NN classifier to identify the gender, age, and educational environment of a user. Experimental evaluation involving 100 participants grouped in gender (2 groups), age (4 groups), and educational environment (10 groups), yielded accuracies of 82.2%, 75.4% and 68.8%, respectively.

2.2.4 Authorship Verification

Authorship verification follows typical biometric verification process, where the identity of an author is verified through one-to-one matching. Some researchers have investigated authorship verification as a similarity detection issue, where the problem consists of determining the degree of similarity given two pieces of text, by measuring the distance between them. Other researchers have investigated this issue as a one or two-class problem, with one class composed by documents written by the author, and a second class composed by documents written by other authors.

As part of this previous work, Koppel and Schler [66] introduced a technique named “unmasking” where they quantify the dissimilarity between the sample document produced by the suspect and that of other users (i.e. imposters). They used SVM with linear kernel and addressed the authorship verification as a one-class classification problem. The dataset was composed by 10 authors, where 21 English books were split in blocks of 500 words. Although the overall accuracy was 95.7% when analysing the feature set composed by the 250 most frequent words, they concluded that the use of negative examples could improve the results. In addition, the proposed approach can provide trustable results only for documents of at least 500 words long, which is not realistic in the case of online verification.

Iqbal et al. [56] experimented with two different approaches. The first approach conducts verification using classification; three different classifiers are investigated, namely, Adaboost.M1, Bayesian Network, and Discriminative Multinomial Naive Bayes (DMNB). The second approach conducts verification by regression; three different classifiers were studied including linear regression, SVM with Sequential Minimum Optimization (SMO), and SVM with RBF kernel. The feature set was composed of 292 features, which included lexical (collected either in terms of characters or words), syntactic (punctuation and function words), idiosyncratic (spelling and grammatical mistakes) and content-specific (keywords commonly found in a specific domain). Experimental evaluation of the proposed approach using the Enron e-mail corpus and by analysing 200 e-mails per author, yielded EER ranging from 17.1% to 22.4%.

Canales and colleagues [23] combined stylometry and keystroke dynamics analysis for the purpose of authenticating online test takers, and used k-NN algorithm for classification. The extracted features consisted of 82 stylistic features including 49 character-based, 13 word-based, and 20 syntactic features. Experimental evaluation involved 40 students with sample document size ranging between 1710 and 70,300 characters, yielding as performances (FRR=20.25%, FAR=4.18%) and (FRR= 93.46%, FRR=4.84%) when using separately keystroke and stylometry, respectively. The combination of both types of features yielded EER of 30%. The feature set included character-based, word-based, and syntactic features. They concluded that the feature set must be extended and certain type of punctuations may not necessarily represent the style of students when taking online exams.

Chen and Hao [26] proposed to measure the similarity from email messages by mining frequent patterns. A frequent pattern is defined as the combination of the most frequent features that occur in the emails from a target user. The proposed feature set included 40 lexical, 76 syntactic, 25 content specific, and 9 structural features. They used PCA, k-NN and SVM as classifiers and evaluated the proposed approach using a subset of the Enron dataset involving 40 authors. Experimental evaluation yielded 84% and 89% classification accuracy rates for 10 and 15 short e-mails, respectively.

The authorship track organized yearly at the PAN (Uncovering Plagiarism, Authorship, and Social Software Misuse) competition focused in 2013 and 2014 (i.e. PAN-2013 and PAN-2014) on authorship verification [60, 95]. All teams competed in two categories: intrinsic verification (as one-class problem) and extrinsic verification (as two-class problem). The evaluation dataset was composed by a set of d documents per author for training and a single document per author for testing. The PAN-2014 corpus contains essays, reviews, novels, and articles written in Dutch, English, Greek, and Spanish languages [95]. The average text length is 1,415 words per document, as opposed to e-mail and tweets, which are very short texts and quite poorly structured or written. Most of the teams used simple character n-gram and word-based features, and a shallow architecture for classification. The winners of both PAN-2013 and PAN-2014 competitions on authorship verification were modifications of the "impostors" method proposed by Koppel and Winter [69].

The "impostors" method [69], is an unsupervised method for authorship verification that was evaluated using a dataset consisting of 500 blog pairs. Koppel and Winter analyzed fragments or chunks of blogs consisting of 500 words and extracted as features the 100,000 most frequent character 4-grams. The proposed method consists of transforming the authorship problem from a one-class to a multi-class classification problem by adding additional authors from external sources (e.g. the Web). The experimental evaluation yielded a classification rate of 87.4% for the blog dataset.

2.2.5 Discussion

The architecture of stylometry-based authorship analysis framework follows the classic biometric process and system architecture outlined earlier. The process starts by extracting some features from authors' documents during the enrolment phase and creating a user profile. The matching phase consists of determining whether or not an anonymous document belong to a specific author. The matching phase in authorship identification is based on one-to-many classification, whereas authorship verification is based on one-to-one classification.

Table 2.3 shows comparative performances, block sizes and, validation population sizes for existing stylometry studies from the literature. Previous work in authorship verification used sample population size varying from 2 to 40 authors, achieving accuracy higher than 95% [31, 108]. There are also previous research in authorship attribution with population sizes of 10,000 and 100,000, but the accuracies are only 46% and 20%, respectively [68, 81]. The increase in the number of authors tend to decrease significantly the accuracy.

The block size refers to the size of the analyzed text. Some studies provide the block size in number of words and other in number of characters. According to Sanderson and Guenter the average word length is about 5.6 characters. Block sizes varying from 250 characters to 70,300 characters have been used in the literature [18, 23]. For example, Cheng et al. [28] grouped and analyzed messages with 50, 100 and 200 characters per e-mail. Koppel et al. [68] used 500 words in order to determine the authorship. Sanderson and Guenter [90] have shown promising results with blocks of texts of 500 characters. Kucukyilmaz et al. [73] concatenated multiple

Table 2.3: Comparative performances, block sizes and, population sizes for stylometry studies

Type	Ref	Sample Size	Block Size	Number of Features	Technique	Accuracy* (%)	EER (%)
Attribution	[2]	100	277 w	L(25065), Sy(2766), A(128)	PCA	83.10	- -
	[25]	10	200 w	L(1), Sy(10)	Discriminant Function Analysis (DFA)	95.70	- -
	[31]	2 - 4	60,000 w	Se	Synonym-based features through statistical classification	93.8 - 97.8	- -
	[40]	3	20 sentences	L(28820), Sy(4117), Se(1896)	SVM	87.63	- -
	[42]	3	200 w	L, Sy, and A (400)	SVM and C4.5	69 -83	- -
	[49]	87	287 w	L , Sy(8)	Logic Fuzzy	50 - 60	- -
	[53]	3 - 10	200 w	L(82), Sy(311), A(26)	Expectation Maximization (EM), and k-NN	80 - 90	- -
	[54]	4 - 20	300 w	L(105), Sy(159), Se(10), A(28)	Frequent pattern	69.75 - 88.37	- -
	[55]	6 - 10	200 w	L, Sy, A	Frequent pattern	77 - 80.5	- -
	[76]	20	169 w	L(87), Sy(158), Se(11), A(14)	SVM	99.01	- -
	[83]	20	600 w	Sy(171)	Prediction by Partial Matching (PPM)	84.30	- -
	[90]	50	500 ch	L	Markov chains	- -	8.08 - 30.88
	[68]	10,000	500 w	L (n-gram)	k-NN (cosine similarity)	46	- -
	[81]	100,000	335 w	L(95) , Sy(1093)	k-NN, Naive Bayes (NB), and SVM	20	- -
Characterization	[1]	5	76 w	L(79), Sy (262), Se(15), A(62)	C4.5 decision tree and SVM	90.1 - 97	- -
	[27]	108	50 - 200 w	L(130), Sy(402) , A(13)	SVM, Bayesian logistic regression, and AdaBoost decision tree	73 - 82.23	- -
	[28]	114	50 - 200 w	L(130), Sy(402), A(13)	Decision Tree, SVM	80.08 - 82.20	- -
	[32]	325	50 - 200 w	L(69), Sy(122), A(30)	SVM	70.20	- -
	[54]	4 - 20	300 w	L(105), Sy(159), A(15), Se(23)	Frequent Pattern	39.13 - 60.44	- -
	[73]	100	300 w	L(89), Sy(119), A(3)	k-NN, NB, Patient rule induction method, SVM	39.0 - 99.70	- -
	[87]	10 - 40	450 w	L	Probabilistic Context-Free Grammar (PCFG)	68.3 - 91.5	- -
Verification	[23]	40	1710 - 70300 ch	L(62), Sy(20)	k-NN	- -	30
	[26]	25 - 40	30 - 50 w	L(40), Sy(76), Se(25), A(9)	SVM	83.90 - 88.31	
	[43]	8	628 - 1342 w	L(100K), Sy(900K)	Weighted Probability Distribution Voting (WPDV)	- -	3
	[66]	10	500 w	L(250)	SVM	95.70	- -
	[72]	29	2400 w	L(40)	Linear Discriminant Analysis (LDA)	- -	22

* The accuracy is measured by the percentage of correctly matched authors in the testing set.
(L) = Lexical, (Sy) = Syntactic, (Se) = Semantic, (A) = Application, ch = characters, w = words

chat messages into a single long message consisting of 3,000 words.

The accuracy tends to degrade when the block size becomes smaller [90]. Smaller block size means shorter authentication delay, which is important for CA. Therefore, there is a need to investigate even shorter messages to be able to cover a broader range of online messages such as twitter feeds and text messages. However, attempting to reduce at the same time the block size and verification error rates is a difficult task in the sense that these attributes are loosely related to each other.

Most of the previous work on stylometry have included a combination of lexical, semantic, syntactic, and application-specific features. As we can see in Table 2.3, some studies used over a thousand stylistic features [2, 40]. However, there is no consensus among researchers regarding what is the best set of features. Stylometry features are discussed in details in the next chapter.

Regardless of the approach used for investigation, all proposed models have in common a total reliance on shallow machine learning architectures for classification. Examples of shallow classifiers used in stylometry-based authorship analysis include k-Nearest Neighbors (k-NN) [23], Naïve Bayes (NB) [56], Principal Component Analysis (PCA) [26], Linear Discriminant Analysis (LDA) [72], SVM [2, 56, 66], and Decision Tree [1].

Although the performance of stylometry analysis approaches proposed in the literature are promising, there is a need to improve such performance significantly for continuous authentication purpose. The equal error rate could be improved by investigating new machine learning techniques such as Deep Belief Network classifiers, which have been shown to be powerful analysis techniques in handwriting and visual detection of objects.

Another important limitation of many previous stylometry studies is that the performance metrics computed during their evaluations cover only one side of the story, and this is clearly emphasized by Table 2.3. Accuracy is traditionally measured using the following two different types of errors, namely, Type I error (which corresponds to the FRR) and Type II error (which corresponds to the FAR). However, most previous studies calculate only the so-called (classification) accuracy (see Table 2.3) which actually corresponds to the true match rate and allows deriving only one type of error, namely, Type II error: $FAR = 1 - Accuracy$. Nothing is said about Type I error in these studies, which makes it difficult to judge their real strength in terms of accuracy.

Furthermore, an important issue to achieve a robust CA system is to assess and strengthen the approach against forgeries. Stylometry analysis can be the target of attacks [10]. An adversary having access to writing samples of a user may be able to effectively reproduce many of the existing stylometric features. It is essential to integrate specific mechanisms in the authentication system that would mitigate forgery attacks.

In this dissertation, we tackle the above challenges by developing a new stylometric analysis framework for continuous authentication. The proposed framework relies on authorship verification, which is the centerpiece of any authentication system. Sample texts are decomposed in blocks over which authorship verification occur repeatedly. We investigate short message blocks, which are required to shorten the authentication delays. We also investigate the impact of forgeries by collecting and analyzing forgery data. Finally, we investigate both shallow and deep machine learning classification algorithms, and come up with the conclusion that better results are achieved with the latter category.

2.3 Summary

In this chapter, we provided an overview of biometric authentication and discuss related works on authorship analysis using stylometry. A number of research works have addressed continuous authentication based on physiological and behavioral biometrics [99]. Physiological biometric technologies used in continuous authentication include face recognition, palmprint verification, sitting postures and electrocardiograms verification. Behavioral biometrics used in continuous authentication include gait, keystroke and mouse dynamics. Stylometry is considered a behavioral biometrics and although many studies have employed stylometric techniques for authorship attribution and characterization, fewer studies have focused on verification, and to our knowledge there is no study on using stylometry for continuous authentication.

A significant number of prior studies have proven the benefit of using linguistic profiling for authorship identification and verification. Despite significant progress in identifying an author among a few candidates (eg. 3 to 10), it is still challenging to identify an author when we have a large number of candidates or when the text is short such as an e-mail or an online chat message.

We propose to use stylometry-based authorship verification for continuous authentication by analyzing short messages corresponding to reduced authentication windows. In the next chapter we will discuss in more detail our experiment methods and datasets.

Chapter 3

Experiment Method and Datasets

In this chapter, we present the methodology used for the experimental evaluation of the stylometric analysis approaches introduced in subsequent chapters. We also give an outline of the evaluation metrics and the datasets used in our experiments.

This chapter is organized as follows. Section 3.1 presents our proposed authorship verification methodology. Section 3.2 describes the evaluation datasets. Section 3.3 describes data pre-processing steps. Section 3.4 summarizes the metrics used to evaluate the proposed approaches. Finally, we summarize the chapter in Section 3.5.

3.1 Methodology

Our authorship verification methodology is structured around the steps and tasks of a typical pattern recognition process, as shown in Figure 3.1. While traditional documents are very well structured and large in size providing several stylometric features, short online documents (e.g., e-mails and tweets) typically consist of a few paragraphs, wrote quickly and often with syntactic and grammatical errors. In the proposed approach, all the sample texts used to build a given author profile are grouped into a single document. This single document is decomposed into consecutive blocks of short texts over which (continuous) authentication decisions happen. Predictive features (n-best) are extracted from each block of text creating training and testing instances.

The classification model consists of a collection of profiles generated separately for individual users. The proposed system operates in two modes: enrolment and verification. Based on sample training data, the enrolment process computes the behavioral profile of the user.

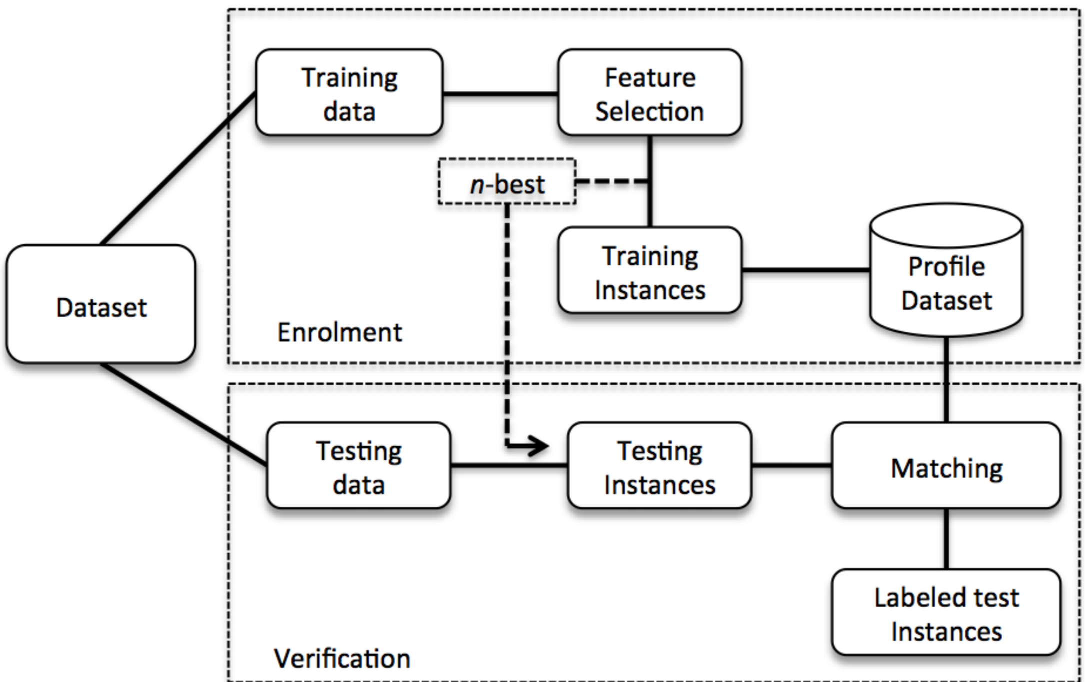


Figure 3.1: Overview of the proposed authorship verification methodology

The verification process compares unseen block of texts (testing data) against the model or profile associated with an individual (i.e. 1-to-1 identity matching) and then categorizes the block of text as genuine or impostor. In addition, the proposed system addresses the authorship verification as a two-class classification problem. The first class is composed by (positive) samples from the author, whereas the second class (negative) is composed by samples from other authors.

3.2 Datasets

In order to validate our work, we use three different datasets. The first dataset is based on a real-life dataset from Enron e-mail corpus[1], while the second dataset is based on a micro-messages corpus based on Twitter feeds [2]. The third dataset is a forgery corpus that was created by simulating forgery attacks against a subset of users' from the Twitter dataset. The three datasets are described in details in the remaining of this section.

3.2.1 E-mail Dataset

The Enron corpus[3] is a large set of email messages from Enron's employees. Enron was an energy company (located in Houston, Texas) that was bankrupt in 2001 due to white collar fraud. The company email database was made public by the Federal Energy Regulatory Commission during the fraud investigation. The raw version of the database contains 619,446 messages belonging to 158 users. However, Klimt and Yang [64] cleaned the corpus by removing some folders that appeared not to be related directly to the users. Therefore, the version used in this dissertation contains more than 200,000 messages belonging to 150 users with an average of 757 messages per user. The e-mails are plain texts and cover various topics ranging from business communications to technical reports and personal chats.

3.2.2 Micro Messages Dataset

Twitter is a microblogging service that allows authors to post messages called "tweets". Each tweet is limited to 140 characters and sometimes express opinions about different

[1] Available at `http://www.cs.cmu.edu/~enron/`
[2] Available at `http://www.uvic.ca/engineering/ece/isot/datasets/`
[3] available at `http://www.cs.cmu.edu/~enron/`

topics. Twitter has over 200 million active users worldwide, posting 9,100 tweets per second. Registered users can read and post tweets, reply to a tweet, send private messages and re-tweet a message, while unregistered users can only read them. A registered user can follow and be followed by other users. Tweets have also other particularities such as the following:

- the use of emoticons to express sentiments;
- the use of URL shorteners to refer to some external sources;
- the use of a tag "RT" in front of a tweet to indicate that the user is repeating or reposting the same tweet;
- the use of a hashtag "#" to mark and organize tweets according to topics or categories, allowing a topic to be searched easily;
- the use of symbol "@<user>" to link a tweet to a Twitter profile whose user name is "user".

One of the Twitter datasets available for research is the 2011 Text Retrieval Conference (TREC) dataset, which has approximately 16 million tweets. However, the quantity of messages written by the same author is very small and insufficient to run our proposed stylometry experiments, which need at least 28,000 characters per author. Therefore, we decided to create our own dataset by crawling messages of authors from Twitter. Firstly, we need to choose a set of authors with several messages. So, we used a list of the UK's most influential tweeters compiled by Ian Burrell (The Independent newspaper). His methodology to choose the people included help from the social media monitoring group, PeerIndex, with additional input from a panel of experts. We randomly selected 100 names from the 2011[4] and 2012[5] lists and crawled their Twitter accounts.

Our dataset contains on average 3,194 twitter messages with 301,100 characters per author. All tweets in the dataset were posted before November 6th, 2013 (inclusive). The Twitter terms of services forbids third-parties from redistributing Twitter Content[6]. Third-parties are allowed to distribute a set of tweet identifiers (tweet IDs and user IDs). A researcher could use the Twitter REST API to download each tweet in JavaScript Object Notation (JSON) format or to crawl raw HTML pages from the twitter.com site. Although the JSON structure provides several information, we used only the content from the "text" field in our experiments, which characterizes the authorship of a message.

3.2.3 Impostors Dataset

An important issue that we need to address to achieve a robust CA system is to assess and strengthen our approach against forgeries. An adversary having access to writing samples of a user may be able to effectively reproduce many of the existing stylometric features.

In order to assess the robustness of our proposed approach against forgeries attempts, a novel forgery dataset was collected as part of this research. We organized an experiment with volunteers forging tweets. Participants in our experiments consisted of 10 volunteers - including 7 males and 3 females - with ages varying from 23 to 50 years, and different background.

Sample tweets were selected randomly for 10 authors considered as legal users

[4] Available at `http://www.independent.co.uk/news/people/news/the-full-list-the-twitter-100-2215529.html`

[5] Available at `http://www.independent.co.uk/news/people/news/the-twitter-100-the-full-ataglance-list-7467920.html`

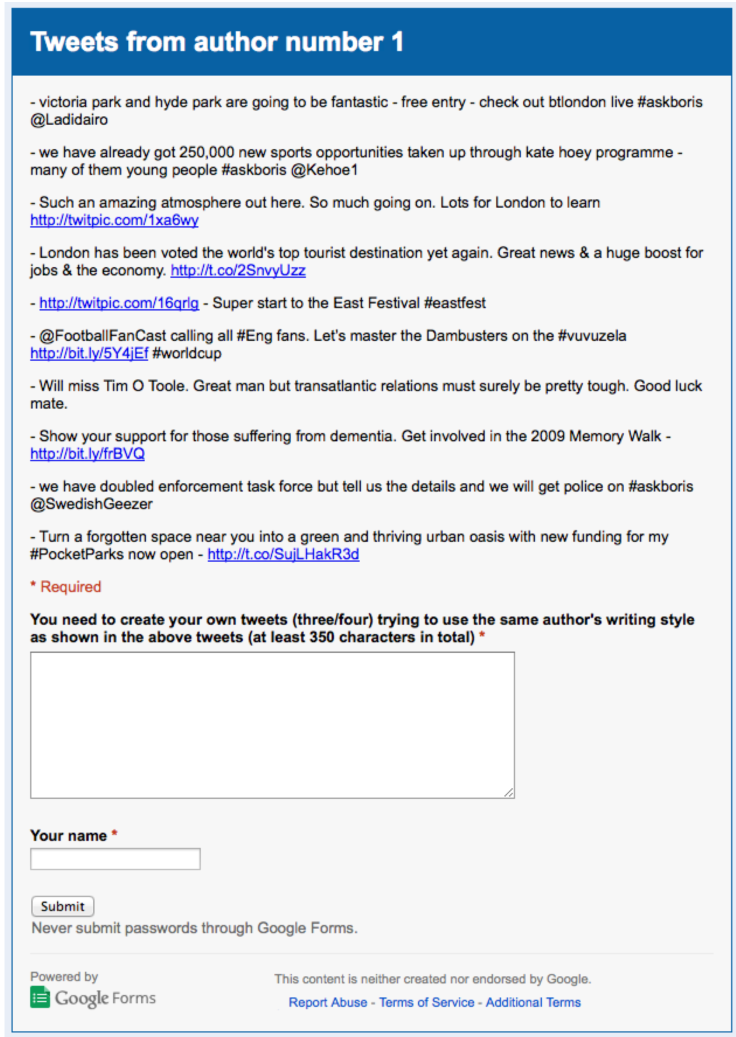
[6] `https://dev.twitter.com/terms/api-terms`

from the Twitter Dataset. Impostor samples were collected through a simple form consisting of two sections. In the first section, tweets from a specific legal user were made available. This allows simulating a scenario where an adversary has access to writing samples. The second section involved two fields, one for participants to enter their names and the other for them to write three or four tweets trying to reproduce the writing style of the legal user. A "submit" button was used to send the sample to the database when completed, as shown in Figure 3.2. The form was sent by email and made available as well online through a web page. We implemented our survey using Google Forms platform. The only restriction imposed was a minimum size of 350 characters per sample spread over 3 to 4 tweets.

We sent to each volunteer a new form with different legal user information, one per every work day. All volunteers were instructed to provide one sample per day. The data was collected over a period of 30 days. We had no control over the way volunteers wrote their tweets. Collected data consisted of an average of 4,253 characters per volunteer spread over 10 attacks.

3.3 Data Preprocessing

Existing datasets consist of a set of candidate users and a set of text samples from these users. The basic assumption for the dataset is that it must contain sufficient information to discriminate different users. In order to have the same canonical form, we will apply preprocessing canonicizer filters to standardize the text, as shown in Figure 3.3. In addition, we have decided to combine all texts from the same author creating a long text and then divide the combined text into smaller blocks of text. Each block of text is treated as a sample over which authentication decision occurs. This approach allows us to simulate repeated authentication windows, which is the



Tweets from author number 1

- victoria park and hyde park are going to be fantastic - free entry - check out btlondon live #askboris @Ladidairo

- we have already got 250,000 new sports opportunities taken up through kate hoey programme - many of them young people #askboris @Kehoe1

- Such an amazing atmosphere out here. So much going on. Lots for London to learn http://twitpic.com/1xa6wy

- London has been voted the world's top tourist destination yet again. Great news & a huge boost for jobs & the economy. http://t.co/2SnvyUzz

- http://twitpic.com/16qrlg - Super start to the East Festival #eastfest

- @FootballFanCast calling all #Eng fans. Let's master the Dambusters on the #vuvuzela http://bit.ly/5Y4jEf #worldcup

- Will miss Tim O Toole. Great man but transatlantic relations must surely be pretty tough. Good luck mate.

- Show your support for those suffering from dementia. Get involved in the 2009 Memory Walk - http://bit.ly/frBVQ

- we have doubled enforcement task force but tell us the details and we will get police on #askboris @SwedishGeezer

- Turn a forgotten space near you into a green and thriving urban oasis with new funding for my #PocketParks now open - http://t.co/SujLHakR3d

* Required

You need to create your own tweets (three/four) trying to use the same author's writing style as shown in the above tweets (at least 350 characters in total) *

Your name *

Submit

Never submit passwords through Google Forms.

Powered by Google Forms

This content is neither created nor endorsed by Google.

Report Abuse - Terms of Service - Additional Terms

Figure 3.2: Screenshot of a form with tweets from an author in the forgery attack experiment

foundation of continuous authentication.

The data was preprocessed in order to normalize e-mail and tweet particularities [21, 35]. In order to obtain the same structural data and improve classification accuracy, we performed several preprocessing steps on the data.

In the Enron corpus, we used only the body of the messages from the e-mails found in the folders “sent” and “sent items” for each user. JavaMail API was used to parse each e-mail and extract the body of the message. Email header information and

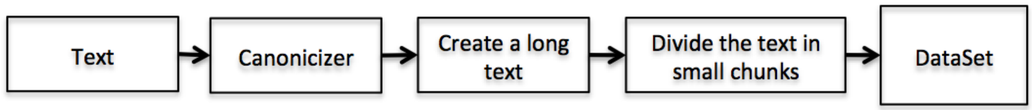


Figure 3.3: Data preprocessing

reply text was not used. All duplicate e-mails were removed. Similarly, we removed all e-mails that contain tables with numbers when the average number of digits per total number of characters was higher than 25%. We also removed reply texts when present and replaced e-mail and web addresses by meta tags “e-mail" and “http", respectively.

In the Twitter and forgery corpuses, we removed all Re-Tweet (RT) posts and all duplicated tweets. Hashtag symbols such as “#word” and the following word were replaced by a meta tag “#hash”; @<user> reference was replaced by meta tag “@cite”; web addresses were replaced by meta tag “http”. We also removed all messages that contain one or more of the following unicode blocks: Arabic, Cyrillic, Devanagari, Hangul-syllables, Bengali, Hebrew, Malaya-lam, Greek, Hiragana, Cherokee, and CJK-unified-ideographs.

In all datasets, we replaced currency by a meta tag “$XX”, percentage by a meta tag “XX%”, date by a meta tag “date”, hours by a meta tag “time”, phone number by a meta tag “phone”, numbers by a meta tag “numb”, information between quotes (e.g., “information”) by a meta tag “quote”, and information between tags (“<information>”) by a meta tag “TAG”. Finally, the document was normalized to printable ASCII, all characters were converted to lower case, and the white space was normalized.

3.4 Evaluation Method

This section outlines various metrics used in the evaluation of biometric systems.

3.4.1 Measures of Classification Performance

During the enrolment mode, we derive the reference profile of a user U based on a training set consisting of samples from the user (i.e. positive samples) and samples from other users (i.e. negative samples) considered as impostors.

Authentication consists of computing the similarity of a sample against the user profile, and comparing the obtained score S against some threshold Th. If the score is greater or equal the to the threshold, the sample will be accepted and considered as genuine. Otherwise, it will be rejected and classified as being from an impostor.

During the above classification process, samples may be wrongly accepted (as genuine) or rejected (as from an impostor). In this context, the accuracy of biometric systems is evaluated primarily in terms of false rejection rate and false acceptance rate.

Given an evaluation dataset, we build a reference profile for each of the p genuine users. To calculate the FRR_U for a specific user U, we compute the number of False Rejections (FR). FR is counted when a positive sample is incorrectly classified as negative. A false rejection will occur if S is below the threshold Th:

$$FR = \begin{cases} 1 \text{ if } S < Th \\ 0, \text{otherwise} \end{cases} \tag{3.4.1}$$

To calculate the FAR_U, we compute the number of False Acceptances (FA). FA is counted when a negative sample is incorrectly classified as belonging to the user U. A false acceptance will occur if S is above the threshold Th:

$$FA = \begin{cases} 1 \text{ if } S \geq Th \\ 0, \text{otherwise} \end{cases} \tag{3.4.2}$$

After all samples from user U are checked, average FRR and FAR are computed as described below:

$$FRR_U = \frac{FR}{\text{all genuine samples}} \times 100 \tag{3.4.3}$$

$$FAR_U = \frac{FA}{\text{all impostor samples}} \times 100 \tag{3.4.4}$$

Then, the overall FRR and FAR over one round are obtained by averaging the individual measures over the entire user population, as shown in Equations 3.4.5 and 3.4.6 respectively.

$$FRR_{round} = \frac{\sum_{i=1}^{p} FRR_i}{p} \tag{3.4.5}$$

$$FAR_{round} = \frac{\sum_{i=1}^{p} FAR_i}{p} \tag{3.4.6}$$

Where i $(1 \leq i \leq p)$ is the user index.

The pair (FAR,FRR) varies according to the selected threshold value. In order to select adequate operating points (or threshold values), different (FAR,FRR) pairs are computed by varying the threshold, and plotting what is known as the Receiver Operating Characteristics (ROC) curve. The ROC curve is a graphical plot of the relation between the FRR and the FAR, as shown in Figure 3.4. Each point on the curve represents the values for FAR and FRR when a specific threshold is used in the calculation. The curve is calculated by varying the threshold value over a specific range.

A common biometric performance metric that can be obtained from the ROC curve is the Equal Error Rate (EER), which corresponds to the operating point

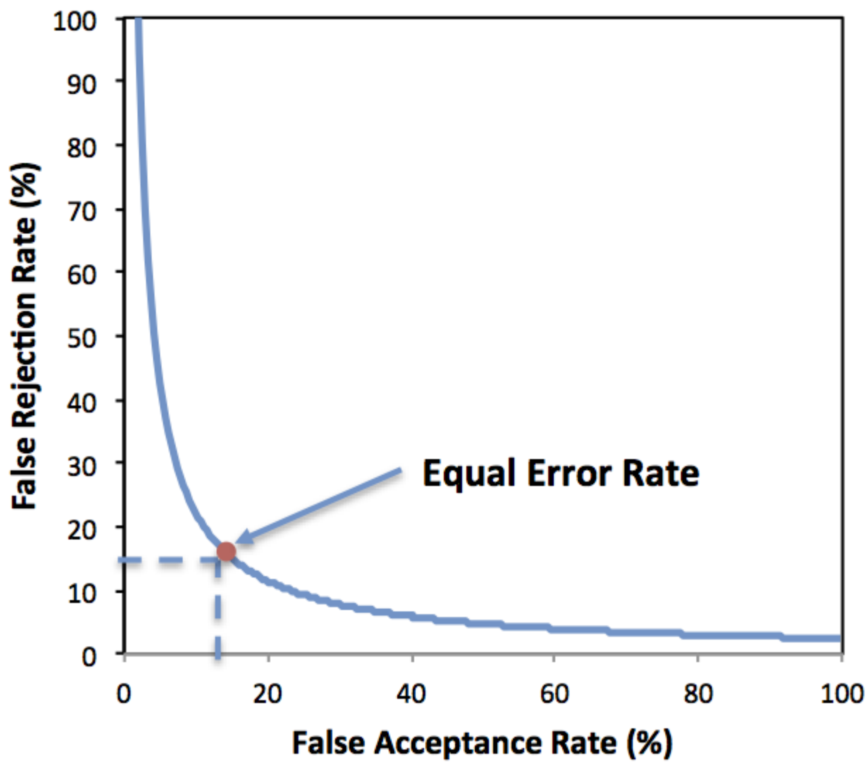


Figure 3.4: Receiver Operating Characteristic curve

where FRR and FAR have the same value. Similarly, another common biometric performance metric, is the Half Total Error Rate (HTER), which is a simple mean of FRR and FAR, as follows:

$$HTER = \frac{FAR + FRR}{2} \tag{3.4.7}$$

To reduce variability, we use cross-validation to determine the performance of our proposed system. Cross validation is a technique for assessing how accurately a classification model will perform in practice, limiting the overfitting problem. One round of cross-validation involves partitioning the dataset in two subsets, one with training samples and another with testing samples. For instance, in a 10-fold cross validation, the dataset is partitioned randomly into 10 equal subsets. Then in each validation round, 9 of the subsets are used for training while the remaining subset is used for testing. This process is repeated ten times by using each time a different subset for testing and the 9 remaining ones for training. Finally, the validation performance results are averaged over the different rounds, as shown in Equations 3.4.8 and 3.4.9 respectively.

$$FRR_{final} = \frac{\sum_{round=1}^{10} FRR_{round}}{10} \tag{3.4.8}$$

$$FAR_{final} = \frac{\sum_{round=1}^{10} FAR_{round}}{10} \tag{3.4.9}$$

3.4.2 Confidence Interval

Providing the above performance metrics alone is not enough to measure the reliability of a biometric system. Instead, confidence measures should be associated with the performance measures as well.

In this dissertation, the assessment of confidence of the framework accuracy is based on the method proposed by Bengio and Mariethoz [12]. According to these authors, standard statistical tests cannot be used to measure the statistical significance of person authentication models, because several of the performance metrics used to assess those models, such as EER or HTER are aggregates of two measures (e.g., FAR and FRR). As a result, they proposed a new method to calculate the confidence interval (CI) by approximating the distribution of the number of errors to a normal distribution with standard deviation σ.

The CI around an HTER is computed as $HTER \pm E$, where E is the margin of error. E is defined as follows:

$$E = \sigma \times Z_{\delta/2} \tag{3.4.10}$$

Where $Z_{\delta/2}$ is the confidence coefficient, δ is the confidence level, and σ is the standard deviation.

3.5 Summary

In this chapter, we give an outline of the experimental methodology used in our work. We also introduce the different datasets used for experimental evaluation, and present standard metrics used to assess the performance of biometric systems.

In the next chapter we will discuss the most common writing characteristics used to create a profile that represents the style of an author, and introduce feature selection technique to reduce large feature space and eliminate redundant features.

Chapter 4

Feature Space

Stylometry consists of the quantification of the writing style elements or style markers of a document in order to create a writeprint that represents the style of its author. Stylometric features can be analyzed using lexical, syntactic, semantic and structural techniques. Each one of these techniques has its own unique strengths [97]. In this chapter, we present a summary of sample features proposed in the literature and used in this work, and introduce a new n-gram model that uses a supervised learning technique to derive n-gram features. In addition, we present a strategy to select the best features.

This chapter is organized as follows. Section 4.1 summarizes and discusses existing features described in the literature. Section 4.2 introduces our new n-gram model. Section 4.3 outlines the final feature set. Section 4.4 presents our approach to select the best set of features representing an author's writing style. Finally, Section 4.5 summarises the chapter's content.

4.1 Common Stylometric Features Categories

A specific user can be identified by his relatively consistent writing styles. According to Iqbal et al. [55] the writing style of a user "contains characteristics of words usage, words sequence, composition and layouts, common spelling and grammatical mistakes, vocabulary richness, hyphenation and punctuation". A broad categorization of stylometric features include the following groups of features: lexical, structural, semantic, and application-specific. A brief description of each feature subset is given below.

4.1.1 Lexical Features

Lexical features are related to the words or vocabulary of a language. Lexical analysis is used to decompose a text into a single atomic unit of language called token. Each token can be a word or a character [6]. While earlier studies used a set of 100 frequently used words to determine the author of a document [20], recent studies have used more than 1000 frequently used words to represent the style of an author [49]. Lexical features encompass not only the frequency of characters or words found in a text, but also vocabulary richness, sentence/line length, word length distribution, n-grams and lexical errors [23, 94]. For the sake of simplicity, lexical analysis can be divided into character and word levels.

Character-level measures the frequency of characters, which include letters (uppercase and lowercase), digits, and special characters (e.g. '@', '#', '$', '%', '(', ')', '{', '}', etc) as shown in Table 4.1. Another approach used at the character-level is to extract n-grams from a text. N-grams are tokens formed by a contiguous sequence of n items. For example, the sequence "the book is on ..." could be represented as 1-gram (t, h, e, _, b, o, o, k, _, i, s, _, o, n), 2-grams (th, he, e_, _b, bo, oo, ok, k_, _i, is, s_, _o, on) and 3-grams (the, he_, e_b, _bo, boo, ook, ok_, k_i, _is, is_, s_o, _on).

The most frequent n-grams constitute the most important feature for stylistic purposes. Importantly, n-grams are noise tolerant since their representation is not affected dramatically by factors such as misspelling [94]. For example, the misspelled word "mysteri" and the correct word "mystery" produce the same number of 3-grams tokens. Character n-grams have been shown to be more efficient than other features [58,63,84,94]. The best performance in English was achieved with 4-grams tokens [90]. As such, n-grams is one the main features developed in our framework. We introduce in section 4.2 a new n-gram analysis model based on character n-grams.

Another character-based feature is a metric computed from the number of icons occurring in a document. An icon is commonly used to express a writer's mood in online messages and can be used as a stylistic marker [82]. It can be written in a text form, for instance, ":-)", ":o)", or in unicode characters, for instance, ☺ - ☹. We categorized 126 text-based emotion icons in 38 different emotion types (e.g., smiley, laughing, very happy, frown, angry, crying), with an average of 3.31 icons per emotion type. In a unicode character, the range of emoticons vary from code 1F600 to 1F64F with 80 different possible icons. In addition, some authors use miscellaneous symbols in their message, for instance, ♋ or ♫. The range of these symbols in unicode characters is from 2600 to 26FF with 256 different symbols. The metric can be calculated as the average of each symbol per type of emotion.

The earliest lexical-based studies focused only on the analysis of words and sentences. For example, Mendenhall [104] analyzed the word length and the relative frequency of its occurrence creating a characteristics curve. He suggested that a document belongs to an author when the curve remains constant. Zipf [110] analyzed the frequencies of different words and formulated a logarithmic relationship between

Table 4.1: Lexical (Character based) features

Type	**Feature Description**
Frequency	Total number of characters (C)
	Average sentence length in terms of characters
	Average sentence length in terms of vowels (V)
	Ratio of letters to C
	Total number of lower character/C
	Total number of upper characters/C
	Total number of digital characters/C
	Total number of white-space characters/C
	Total number of tab space characters/C
	Number of special characters (%,&,etc.)/C (23 features)
	Ratio of digits to C
	Ratio of vowels (a, e, i, o, u) divided by V
n-grams	Character 2-grams
	Character 3-grams
	Character 4-grams
	Digit 2-gram
	Digit 3-gram
icons	Text based icon (8 groups)
	Unicode - emoticons (code range from 1F600 to 1F64F)
	Unicode - miscellaneous symbols (code range from 2600 to 26FF)

them. This allowed concluding that the most frequent word appears twice as many times as the second most frequent word, three times as many times as the third most frequent word and so on. Yule [106] used sentence-length as a statistical characteristic and Poisson distribution to approximate the words used (known as characteristic K). Williams [103] established that the *log* of the number of words per phrase follows a normal distribution.

Recent studies have measured the average word length and the number of syllables per word [50, 108], applying n-gram frequencies [89], measuring entropy, and calculating the total number of words longer than 6 characters and shorter than 3 characters [27]. Table 4.2 shows a list of lexical word-based features proposed in the literature.

Vocabulary richness measures the diversity of vocabulary in a text by quantifying the total number of unique vocabulary, the number of *hapax legomena* (i.e., a word which occurs only once in a text) and the number of *dis legomena* (e.g., dis legomena, tris legomena, ... referring to double, triple, ...) [108].. This metric is computed by dividing the total number of unique vocabulary, *hapax legomena* or *dis legomena* by the total number of tokens (each token is a word).

Another type of word-based features is based on measures of functional words [27, 42, 59]. Functional words are used to express a grammatical relationship with other words. Examples of functional words include articles, pronouns, conjunctions, auxiliary verbs, interjections, adoptions, particles, and expletives. Clark et al. [31] and Kucukyilmaz et al. [73] eliminate content-independent terms by ignoring terms that match predefined stop-words (e.g., connectives, conjunctions, and prepositions). This allows reducing the amount of noise.

Word-based features can also be extracted from the analysis of lexical errors. This consists of measuring the number of misspelled words and spelling errors, such as letter omissions, insertions and formatting errors (e.g., a word with two upper-case letters and the remaining letters are lower-case). For example, Abbassi [1] included in his work a list of 5,513 common word misspellings (although he classified this list using an idiosyncratic category), while Inches [52] measured the percentage of out-of-dictionary words (i.e. casted as errors by a standard spell-checker).

4.1.2 Syntactic Features

Syntactic features refer to how phrases are constructed, taking into account the rules underlying the format of phrases as well as how clauses or sentences are put together. Syntactic features can be divided into average number of punctuations, part-of-speech

Table 4.2: Lexical (Word based) features

Type	**Feature Description**
Frequency	Total number of words (N)
	Average sentence length in terms of words
	Long words (more than 6 characters/N)
	Short words (1-3 characters/N)
	Average token length
	Numbers of syllables per word
	Ratio of numbers of characters in words to N
Vocabulary	Hapax legomena and dis legomena
	Number of different words/N
	Yules K
Functional words	Articles, pronouns, conjunctions, auxiliary verbs, interjections, adoptions, particles, expletives and pro-sentences each one divided by N
Lexical errors	Total number of misspelling words / N
	Frequency of misspelling word

(POS), sentence structures, and chunks as illustrated in Table 4.3. Syntactic pattern is an unconscious characteristic and it is considered to be more reliable than lexical information [8]. Punctuation includes single quotes, commas, periods, colons, semi-colons, question marks, exclamation marks, and uncommon marks based on the unicode format (e.g. †, ⁝, ... ∴). Punctuations allow defining boundaries and identifying meaning by splitting a paragraph into sentences and each sentence into various tokens. However, it is not sufficient to analyze only the punctuation of a document, as certain words such as 'Ph.D.' or 'uvic.ca' include punctuation characters too.

The part-of-speech tagging (POS tagging or POST) consists for a simple sentence of categorizing the words (e.g., as an adjective, adverb, noun), and then building a tree. For example, the sentence (*He runs every day on the beach*) could be analyzed as:

PN[*He*] V[*runs*] ADJ[*every*] N[*day*] PN[*on*] ART[*the*] N[*beach*]

where PN, V, ADJ, N, and ART stand for pronoun, verb, adjective, noun, and

Table 4.3: Syntactic features

Type	**Feature Description**
Punctuation	Total number of punctuation (P)
	Single quotes, commas, periods, colons, semi-colons, question marks, multiple question marks, exclamation marks, multiple exclamations marks and ellipsis
Part of Speech	Pronoun (PN), noun (N), adjective (ADJ), verb (V), adverb(ADV), preposition(P), conjunction(C), interjection(ART)
Sentence structure (n-gram)	2-gram (eg. PN, V)
	3-gram (e.g., PN, V, ADJ)
	4-gram (e.g., PN, V, ADJ, N)
	5-gram (e.g., PN, V, ADJ, N, PN)
Chunks	Phrasal verbs
	Polywords
	Collocations
	Institutionalized utterances
	Sentence frame
	Sentence head
	Text frame

article, respectively, according to the TOSCA annotation. The weakness of this type of features is that POS is language-dependent since it relies on a language parser and also could produce some noise due to the unavoidable errors made by the parser [96]. We use a list of functional words[1] as a base of our list. We also use the Stanford Log-linear Part-Of-Speech Tagger to tag the syntactic words according to this list [98]. Characteristics such as part-of-speech frequency and part-of-speech n-gram frequency [52,65,67], sentence structure [62], errors such as sentence fragments and mismatched tense [65], and frequency of word class (e.g., N, V, ADJ) can influence this measure.

Another approach consists of measuring chunks [43], which may consist of a single word or a chain of words that have a meaning. A chunk could include words (e.g., house, car), phrasal verbs (e.g., to get out, to put off), polywords (e.g., by the way,

[1] The list of functional words used in our work is available at `http://www.sequencepublishing.com/academic.html`

inside out), collocations (e.g., crystal clear, motor cyclist), institutionalized utterances (e.g., I'll get it, We'll see, That'll do, If I were you, Would you like a cup of coffee?), idioms (e.g., make a killing, break a leg), sentence frames and heads (e.g., That is not as...as you think, The problem was), and text frames (e.g., In this paper we explore...; Firstly...; Secondly...; Finally ....).

4.1.3 Semantic Features

Semantic features are related to the meaning of the language and involve factors such as the meaning of words, grammatical construction, as well as functional, and semantic relationships [61]. Table 4.4 illustrates examples of semantic features. The meaning of a word could be classified as a simple synonym, an antonym or, in more complex cases, as a hypernym, hyponym, or polysemy. While synonym-based features are related to the frequency of words with the same meaning, antonym features are related to the frequency of words that have opposite meanings [31]. In this case, thesaurus is an important tool to classify words. Hypernym is the general meaning of a word or term, e.g., dog is a hypernym for Shih Tzu. On the other hand, hyponym is the specific meaning of that word, e.g., dog is a hyponym for animal. Polysemy refers to words that have related meanings, e.g., wood and a piece of a tree. WordNet [79] is a lexical database of English words that has been used for semantic-level analysis. For example, McCarthy et al. [77] used WordNet to compute metrics related to hypernymy and polysemy.

Semantic features can also be extracted through dependency analysis. Gamon [40] used a natural language processing tool named NLPWin to build a semantic dependency graph that captures the semantic relationships.

Semantic features can also be analyzed according to functional characteristics. Argamon et al. [7] used four functional lexical features to extract the meaning of a

Table 4.4: Semantic features

Type	Feature Description
Frequency	Synonym
	Antonym
	Hypernym
	Hyponym
	Polysemy
Functional	Conjunction, modality, comment and appraisal

text. These semantic functional characteristics were grouped in terms of conjunction (conjoin clauses, e.g., 'and', 'while'), modality (to qualify events or entities, e.g., 'can' , 'probability'), comment (to express an opinion or reaction, e.g., 'certainly', 'unfortunately') and appraisal (to express an attitude, e.g., 'happy', 'good').

4.1.4 Application-Specific Features

Application specific features can easily be extracted from documents by analyzing structural and content-specific characteristics [26, 33, 42, 108]. Table 4.5 illustrates examples of application-specific features.

Structural characteristics are related to the organization and format of a text and are usually more flexible in online documents such as e-mails. Structural characteristics are classified at the message-level, paragraph-level, or according to the technical structure of the document [2]. Message-level features include greeting and farewell acknowledgements, the position of re-quoted text, the presence of signature, etc. Paragraph-level features include the total number of lines, sentences and paragraphs, and also the average number of sentences, words and characters per paragraph [27, 28]. Features related to the technical structure of the document include font color and size, file extensions, hyperlinks and embedded images [1]. However, some of these characteristics must be analyzed with caution since in organizations complying with standards such as ISO 9001, the use of standardized e-mail formats is a common practice.

Table 4.5: Application-specific features

Type		**Feature Description**
Structural	Message-Level	Number of blank lines/total number of lines
		Average length of non-blank line
		Presence/absence of greeting words
		Presence/absence of farewell words
		Position of re-quoted text
		Presence/absence of signature
	Paragraph-Level	Total number of lines
		Total number of sentences
		Total number of paragraphs
		Average number of characters, words and sentences in a paragraph
		Average number of sentences beginning with upper and lower case
	Technical Structure	Average number of color, size and type of font
		File extension
		Hyperlinks
		Embedded image
Content-specific	Vocabulary	Age
		Gender
		Group

Content-specific features measure the use of certain vocabulary in the text. These features can be useful when identifying the gender, age, or a specific group the author may be part of. For example, within the same group, authors tend to use identical taxonomy in their communication and each generation has its own unique vocabulary [42, 73, 108]. In addition, some approaches measure the use of words indicative of the individual's race, nationality, and even tendency towards certain types of violence [1], as well as the number of gender-specific words [28], and psycho-linguistic cues [27].

However, these features are more useful when the context of the text being analyzed does not vary, avoiding the confounding factor of cross-topic texts [93].

4.2 A New n-Gram Model

In this section, we introduce a new n-gram model that extracts n-gram features using a supervised learning technique. Previous stylometric studies have yielded encouraging results with lexical features, particularly n-grams [23, 49]. Considering that n-gram features are noise tolerant and effective, and online documents (e.g. emails, tweets) are unstructured documents, we focus in our work a particular attention on this type of features. While the approach used so far in the literature for n-gram modeling has consisted of computing n-gram frequency in a given sample document, we propose an innovative approach that analyzes n-grams and their relationship with the training dataset.

4.2.1 N-gram Model

In our proposed n-gram model, we measure the degree of similarity between a block b of characters and the profile of a user U. We analyze the presence or not of a specific n-gram and compute a real-valued similarity metric denoted $r_U(b, m)$. We consider two different modes of calculation for n-gram represented by the binary variable m called *mode* (unique n-grams[2] ($m = 0$) and all n-grams ($m = 1$)), and by considering all n-grams with frequency equal or higher than some number f.

Our model consists of a collection of profiles generated separately for individual users. The training phase, during which the user profile is built, involves two steps. During the first step, the user profile is derived by extracting n-grams from sample

[2]Unique n-gram refers to *n-gram* type, i.e., duplicated n-grams are counted as one.

documents. During the second step, a user specific threshold is computed and used later in the verification phase.

As illustrated by Figure 4.1, given a user U, we divide randomly her training samples into two subsets, denoted $T(f)_1^U$ and T_2^U, allocating 2/3 of the training samples to subset $T(f)_1^U$ and 1/3 of the training data to subset T_2^U. We divide T_2^U into p blocks of characters of equal size: $b_1^U, ..., b_p^U$.

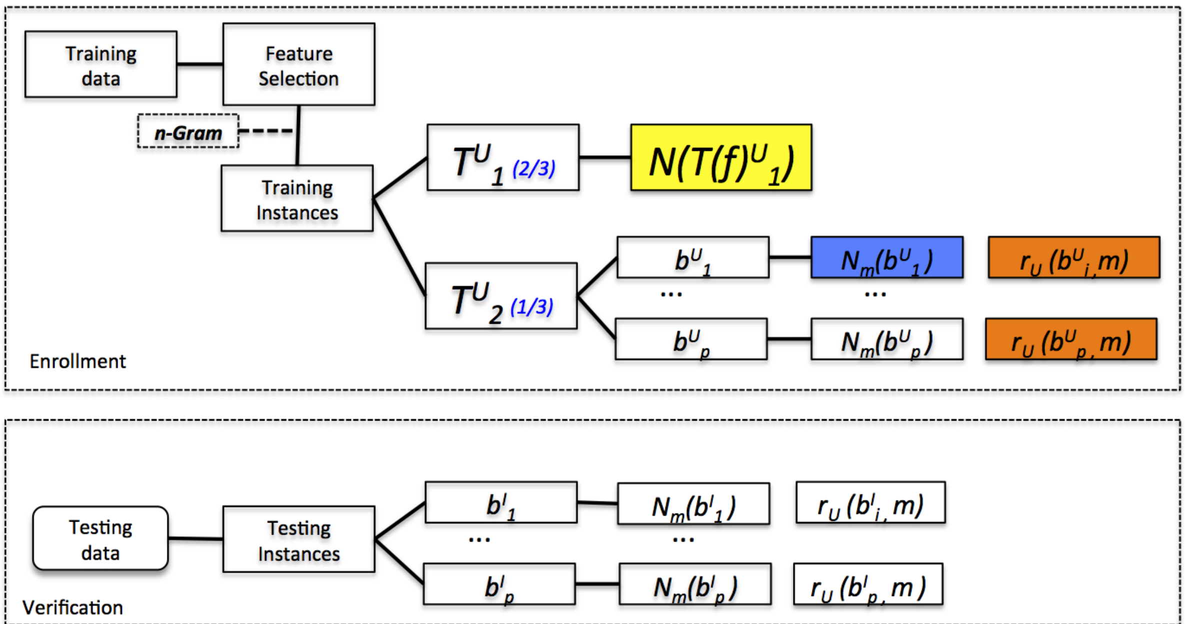


Figure 4.1: Sketch of the new n-gram modeling approach

Given two users U and I, let $r_U(b_i^I, m)$ denote the percentage of unique n-grams shared by block b_i^I (of user I) and (training set) T_1^U, giving:

$$r_U(b_i^I, m) = \frac{|N_m(b_i^I) \cap N(T(f)_1^U)|}{|N_m(b_i^I)|} \tag{4.2.1}$$

Where $N(T(f)_1^U)$ denote the set of all unique n-grams occurring in $T(f)_1^U$ with frequency f, $N_m(b_i^U)$ denote the set of all unique n-grams occurring in b_i^U (for $m = 0$) or the set of all n-grams occurring in b_i^U (for $m = 1$), and $|X|$ denotes the cardinality of set X.

Given a user U, our model approximates the actual (but unknown) distribution of the ratios $(r_U(b_1^U, m), ..., r_U(b_p^U, m))$ (extracted from T_2^U) by computing the sample mean denoted μ_U and the sample variance σ_U^2 during the training.

A block b is said to be a genuine sample of user U if and only if $|r_U(b, m)| \geq (\epsilon_U + \gamma)$, where ϵ_U is a specific threshold for user U, and γ is a predefined constant:

$$\begin{cases} \text{genuine or 1 if } |r_U(b, m)| \geq (\epsilon_U + \gamma) \\ 0, \text{otherwise} \end{cases} \tag{4.2.2}$$

We derive the value of ϵ_U for user U using a supervised learning technique outlined by $Algorithm$ 1 when given training samples from other users $I_1, ..., I_k$ $(I_i \neq U)$. Let *up* and *down* be local variables (in the algorithm) used to verify whether the difference between FRR and FAR is increasing or decreasing. The threshold is initialized (i.e. $\epsilon_U = \mu_U - (\sigma_U/2)$), and then varied incrementally by minimizing the difference between FRR and FAR values for the user, the goal being to obtain an operating point that is as close as possible to the EER (i.e. $FRR = FAR$) for $\gamma = 0$.

In each iteration, the FRR and FAR for user U denoted FRR_U and FAR_U, respectively, are calculated for the current values of ϵ_U and γ. Let δ be a local variable (in the algorithm) that denote the increment/decrement for the ϵ_U value. If $(FRR_U - FAR_U) > 0$, a true value is assigned to the variable *down* and the threshold is decreased by δ. If $(FRR_U - FAR_U) < 0$, a true value is assigned to the variable *up* and the threshold is increased by δ. Finally, we test if *up* and *down* are true, which means that a local optimum was found. In this case, the values of *up* and *down* are reset to false and δ is divided by 10. This process is repeated until δ is lower than 0.0001.

$Algorithm$ 2 returns the FAR and FRR values for a user U given some training data, a user-specific threshold value, and some constant value assigned to γ.

Algorithm 1: Threshold calculation for a given user

```
/* U a user for whom the threshold is being calculated            */
/* I_1, ..., I_k: a set of other users (I_i ≠ U)                  */
/* ε_U: threshold computed for user U                             */
Input: Training data for U, I_1, ..., I_k
Output: ε_U
begin
    up ← false;
    down ← false;
    δ ← 1;
    ε_U ← μ_U − (σ_U/2);
    while δ > 0.0001 do
        /* Calculating FAR and FRR for user U                     */
        FRR_U, FAR_U = getFRRFAR(ε_U, γ, U, I_1, ..., I_k); /* Minimizing the
            difference between FAR and FRR                        */
        if (FRR_U − FAR_U) > 0 then
            down ← true;
            ε_U ← ε_U − δ;
        else if (FRR_U − FAR_U) < 0 then
            up ← true;
            ε_U ← ε_U + δ;
        else
            return ε_U;
        end
        if (up & down) then
            up ← false;
            down ← false;
            δ ← δ/10;
        end
    end
    return ε_U;
end
```

Algorithm 2: FAR and FRR calculation for a given user

Input: getFRRFAR(ϵ_U, γ, Training data for $U, I_1, ..., I_k$)
Output: (FAR_U, FRR_U)

begin
    `/* Calculating FRR for user U */`
    for $i \to 1$ **to** p **do**
        $FR \leftarrow 0;$
        if $r_U(b_i^U)) < (\epsilon_U + \gamma)$ **then**
            $FR \leftarrow FR + 1;$
        end
    end
    $FRR_U \leftarrow \frac{FR}{p};$
    `/* Calculating FAR for user U */`
    for $i \to 1$ **to** k **do**
        for $j \to 1$ **to** n **do**
            $FA \leftarrow 0;$
            if $r_U(b_j^{I_i}) \geq (\epsilon_U + \gamma)$ **then**
                $FA \leftarrow FA + 1;$
            end
        end
    end
    $FAR_U \leftarrow \frac{FA}{p \times k};$
    return $(FAR_U, FRR_U);$
end

4.2.2 Model Evaluation

In order to evaluate the effectiveness of our new n-gram model, we performed several experiments using the Enron e-mail corpus introduced earlier. For classification, we use a simple threshold scheme considered as baseline model. We compute for each user U a corresponding profile by using their training data and training data from other users considered as impostors. This allows computing the acceptance threshold ϵ_U for user U as explained before. A given block b (of characters) is considered to belong to an hypothesized genuine user U when the ratio $|r_U(b)|$ is greater than $\epsilon_U+\gamma$, where γ is a predefined constant and ϵ_U is the user specific threshold.

Evaluation Method

We performed several preprocessing steps to the data as described in Section 3.3. After the preprocessing phase, the dataset was reduced from 150 authors to sets of 107, 92 and 87 authors to ensure that only streams of text with 12,500, 18,750 and 25,000 characters were used in our analysis, respectively. As shown in Figure 4.2, all the sample e-mails used to build a given author profile were grouped into a single document and subsequently divided into small blocks.

We assessed experimentally the effectiveness of our approach through a 10-fold cross-validation test. We randomly sorted the dataset, and allocated in each (validation) round 90% of the dataset for training and the remaining 10% for testing. The 90% training data allocated to a given user U was further divided as follows: 2/3 of the training data allocated to subset T_1^U and 1/3 of the data for subset T_2^U, respectively. The 10% test data for user U was divided in p blocks of equal size s. We tested two different block sizes, $s = 250$ and $s = 500$ characters, respectively. The number of blocks per user p varied from 25 to 100. In addition, we investigated separately n-grams of sizes (n=) 3, 4, 5, and 6, for each of these analyses yielding in

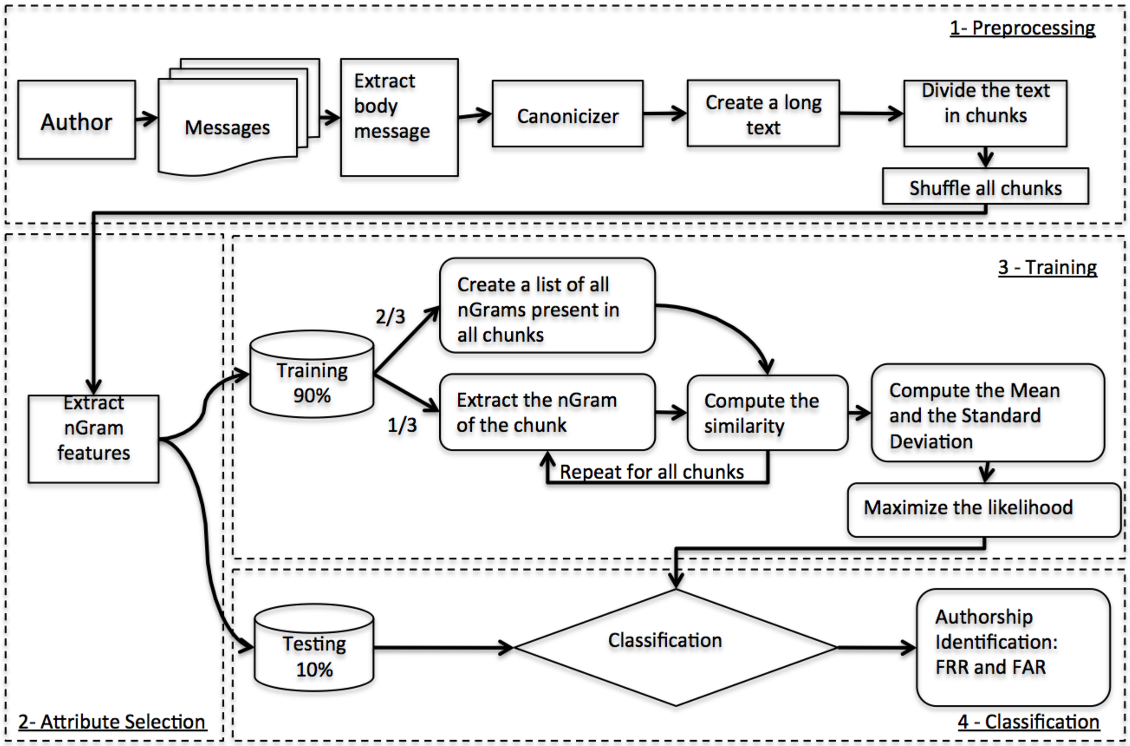


Figure 4.2: The n-gram evaluation method during the enrolment and verification phases.

total 24 different experiments. Table 4.6 shows the configuration of our experiments.

Our experiments cover three different values for the frequency f (i.e. $f = \{0, 1, 2\}$) and two different values for the mode of calculation of the n-grams (i.e. $m = \{0, 1\}$).

We compute the FRR for user U by comparing each of the blocks from her test data against her profile. The FAR is computed by comparing each of the test blocks from the other users (i.e. the impostors) against the profile of user U. By repeating the above process for each of the users, we compute the overall FAR and FRR by averaging the individual measures.

Table 4.6: Configuration of experiments

Experiment configuration #	Number of Users (k)	Number of blocks per author (p)	Block size (s)
1	107	50	250
2	92	75	
3	87	100	
4	107	25	500
5	92	37	
6	87	50	

Evaluation Results

Table 4.7 shows the overall FRR and FAR for the 24 experiments, with the following constants: $\gamma = 0$, $f = 0$, and $m = 0$. It can be noted that the accuracy decreases not only when the number of authors increases, but also when the number of blocks per user p and the block size s decrease.

Experiments using 5-grams achieve better results than those using 3, 4, and 6-grams for large number of blocks per user and large block size. Overall, the best results are achieved in experiment 6, with 87 authors, 50 blocks per user, and a block size of 500 characters (FRR=14.71%, FAR=13.93%).

Table 4.7: Performance results for the different experiments ($\gamma = 0$, $f = 0$, $m = 0$)

No.	*3-gram*		*4-gram*		*5-gram*		*6-gram*	
	FRR	FAR	FRR	FAR	FRR	FAR	FRR	FAR
1	24.85	28.61	22.05	24.09	24.11	20.50	23.18	20.94
2	26.76	26.82	23.64	21.68	25.13	19.39	20.65	20.54
3	24.82	28.15	23.56	21.15	17.24	20.39	20.57	19.56
4	26.47	22.70	23.67	17.81	23.98	16.29	18.71	17.33
5	23.36	21.81	18.75	18.01	18.20	15.40	18.39	15.95
6	22.29	22.21	19.77	16.11	**14.71**	**13.93**	16.78	14.89

Based on configuration 6 (which yields the best results), we assess the impact of the frequency f and mode m on the system performance, by varying the frequency from 0 to 2 and the mode between 0 and 1. Table 4.8 lists the obtained results.

Table 4.8: Performance results by varying f and m for experiment number 6 ($\gamma = 0$)

f	m	*3-gram*		*4-gram*		*5-gram*		*6-gram*	
		FRR	FAR	FRR	FAR	FRR	FAR	FRR	FAR
0	0	22.29	22.21	19.77	16.11	14.71	13.93	16.78	14.89
	1	21.83	22.16	19.08	16.02	15.40	13.90	16.32	14.77
1	0	21.60	22.93	21.83	17.09	19.54	15.45	19.07	17.43
	1	21.60	22.87	22.75	17.30	20.68	15.46	19.09	17.53
2	0	23.21	22.20	21.83	18.11	21.37	16.38	20.46	18.67
	1	22.75	22.78	21.60	18.47	22.98	16.46	20.23	19.12

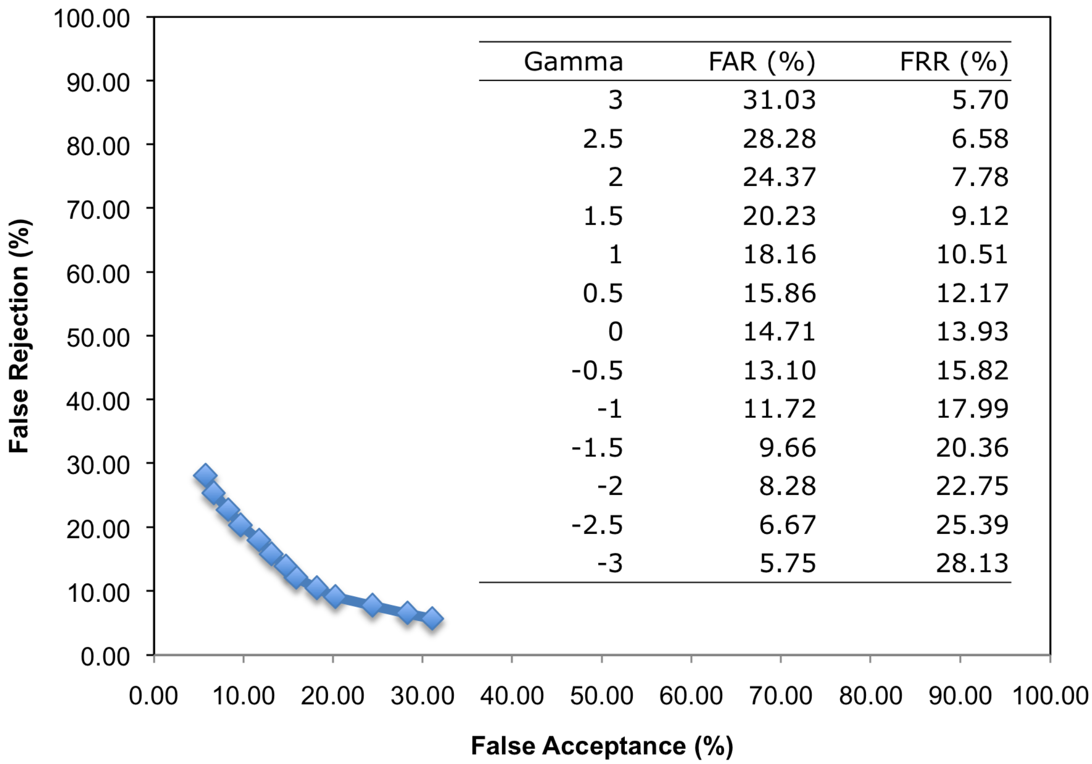
Figure 4.3 depicts the ROC curve for experiment configuration # 6 (from Table 4.6) using 5-gram, $f = 0$, and $m = 0$. The curve illustrates the relationship between the FRR and FAR for different values of γ. The EER was estimated as 14.35% and achieved when $\gamma = -0.25$.

4.2.3 Comparison with a Baseline Method

In this section, we compare our n-gram model with the traditional n-gram model. In the traditional n-gram model, the feature vector is composed of the respective frequencies of the different space-free character n-gram in a document [69], which means that each n-gram is an individual feature.

Baseline approaches from the literature for n-gram modeling use metrics such as euclidean distance or cosine similarity [24, 81]. In this section, we use the Euclidean distance as baseline model.

Given a user U, we divide her training dataset into two subsets, T_1^U and T_2^U, as indicated above. We calculate the similarity between a block b of characters and the training dataset, by generating a cluster from T_1^U with centroid c^U, computed as a single mean vector. The euclidean distance between a block b and c^U is given by:



Gamma	FAR (%)	FRR (%)
3	31.03	5.70
2.5	28.28	6.58
2	24.37	7.78
1.5	20.23	9.12
1	18.16	10.51
0.5	15.86	12.17
0	14.71	13.93
-0.5	13.10	15.82
-1	11.72	17.99
-1.5	9.66	20.36
-2	8.28	22.75
-2.5	6.67	25.39
-3	5.75	28.13

Figure 4.3: Receiver Operating Characteristic curve for experiment configuration #6 using 5-*gram* and sample performance values for different values of γ

$$|\overrightarrow{b} - \overrightarrow{c^U}| = \sqrt{\sum_{i=1}^{n} (b_i - c_i^U)^2} \tag{4.2.3}$$

Where n is the number of features.

We compute a user-specific threshold for user U, denoted t_U as the average of the distances between each of the blocks b from T_2^U and the centroid c^U. A block b is said to be a genuine sample of user U if and only if $|\overrightarrow{b} - \overrightarrow{c^U}| \leq t_U$, which means that it is closer to the centroid c^U.

The baseline experiment was conducted using the Enron dataset involving 87 authors, block size of 500 characters per user, 50 blocks per author, and extracting 5-grams characters as the feature set, since this configuration achieved better result than other configurations in our previous evaluation experiments. Experiment using 5-grams for the traditional model yielded EER of 21.26%. This indicates that our proposed n-gram model (which achieves $EER = 14.35\%$) outperforms the traditional baseline model.

4.2.4 Derived Features

In addition to the real-valued similarity metric introduced above, we define a new binary similarity metric $d_U(b)$ based on $r_U(b,m)$ as follows:

$$\begin{cases} d_U(b) = 1 \text{ if } |r_U(b,m)| \geq \epsilon_U \\ d_U(b) = 0, \text{otherwise} \end{cases} \tag{4.2.4}$$

Where ϵ_U is a user-specific threshold derived from the training data, using *Algorithm* 1.

In addition to n-grams, for each test block b, we derive 2 new features corresponding to $r_U(b,m)$ and $d_U(b)$. In this study, we consider only 5-grams and 6-grams[3], and cover two different values for the frequency f (i.e. $f = 1$ and $f = 2$) and for the mode of calculation of the n-grams (i.e. $m = 0$ and $m = 1$). Therefore, the number of new features created from the above n-gram model and incorporated in our final feature set is 2 (for f) $\times 2$ (for m) $\times 2$ (for n-gram types) $\times 2$ (for r_U and d_U) $= 16$.

4.3 Final Feature Set

Although our new n-gram model achieves very encouraging results compared to the existing literature, an EER of 14.35% is still relatively high for continuous authentication, which is the goal of our work. Our approach to improve the accuracy of

[3]5-grams and 6-grams achieve the best results in our previous experiments.

our framework is to investigate more advanced classification models using machine learning, and also to consider a much broader feature space.

Our global feature space consisted initially of a set of existing features selected from the literature, which was then expanded to add adequately several new features. As indicated above, new n-gram features were generated using our new model.

Since Twitter feeds have some peculiarities such as the use of icons and different punctuations, we expanded our feature set by adding a list of icons with 462 symbols representing the writer's mood, and a general punctuation list with 112 different symbols based on the unicode format. In addition, we extended our feature set by adding the fifty most frequently used words per author. Our final feature set is composed by 972 features consisting of lexical character (528 features), lexical word (75 features), syntactic (362 features) and application-specific features (7 features). The list of all the features used in this work is shown in Table 4.9.

It is important to note that not all these features may be relevant for all users. Some usage scenarios or datasets may exhibit only a subset of the features, for instance, the Enron (email) dataset does not contain emoticons. In order to identify and keep the most discriminating features, we use adequate feature selection techniques outlined in the next section.

4.4 Features Selection

Over a thousand stylistic features have already been identified and used in the literature along with a wide variety of analysis methods. However, there is no agreement among researchers on which features yield the best results. As a matter of fact, analyzing a large number of features does not necessarily provide the best results, as some features provide very little or no predictive information.

Table 4.9: List of stylometry features used in our work

Feature	Characteristics
Lexical (Character)	
F_1	Number of characters (C)
F_2	Number of lower character/C
F_3	Number of upper characters/C
F_4	Number of white-space characters/C
F_5	Total number of vowels (V)/C
F_6 ... F_{10}	Vowels (a, e, i, o, u) / V
F_{11} ... F_{36}	Alphabets (A-Z) / C
F_{37}	Number of special characters (S) /C
F_{38} ... F_{50}	Special Characters (e.g. '@', '#', '$', '%', '(', ')', '{', '}', etc.) / S
F_{51} ... F_{66}	Character *5* and *6*-grams ($r_U(b)$ and $d_U(b)$) with two different values for the frequency f (i.e. $f = 1$ and $f = 2$) and for the mode of calculation of the n-grams (i.e. $m = 0$ and $m = 1$)
F_{67} ... F_{192}	Text based icon (8 groups)
F_{193} ... F_{272}	Unicode - emoticons
F_{273} ... F_{528}	Unicode - miscellaneous symbols
Lexical (Word)	
F_{529}	Total number of words (N)
F_{530} ... F_{539}	Average sentence length in terms of words /N
F_{540}	Words longer than 6 characters/N
F_{541}	Total number of short words (1-3 characters)/N
F_{542}	Average word length
F_{543}	Average number of syllable per word
F_{544}	Ratio of number of characters in words to N
F_{545} ... F_{550}	Replaced words / N
F_{551} ... F_{600}	The 50 most frequent words per author
F_{601} ... F_{602}	Hapax legomena and dislegomena
F_{603}	Vocabulary richness (Number of unique words/N)
Syntactic	
F_{604}	Total number of punctuation (P)
F_{605} ... F_{612}	Number of punctuation (single quotes, commas, periods, colons, semi-colons, question marks, exclamation marks) / P
F_{613} ... F_{724}	Unicode - General punctuation
F_{725} ... F_{729}	Number of function words (conjunction, determiners, preposition, interjection, and pronouns) / N
F_{730} ... F_{965}	Relative frequency of function word
Application-specific	
F_{966}	Total number of sentences
F_{967}	Total number of paragraphs
F_{968} ... F_{970}	Average number of characters, words and sentences in a block of text
F_{971} ... F_{972}	Average number of sentences beginning with upper case and lower case

Being able to keep only the most discriminating features individually per user allows reducing the size of the data by removing irrelevant attributes and improves the processing time for training and classification. This can be achieved by applying feature selection measures, which allow finding a minimum set of features that represent the original distribution obtained using all the features.

Although feature selection by an expert is a common practice, it is complex and sometime inefficient because it is easy to select irrelevant attributes while omitting important attributes. Other feature selection methods include exhaustive search and probabilistic approach. Exhaustive search is a brute force feature selection method that could evaluate all possible feature combinations, but it is time consuming and impractical. The probabilistic approach is an alternative for speeding up the processing time and selecting optimal subset of features.

Our feature selection approach, depicted by Figure 4.4, builds on previous works by identifying and keeping only the most discriminating features, and by also identifying new sets of relevant features. We derive, from the raw stylometric data, numeric feature vectors that represent term frequencies of each of the selected features. All frequencies are normalized between 0 and 1. Each user has a specific feature set that best represents his writing style. We will present in detail the specific features retained in our final feature space in subsequent chapters.

An ideal feature is expected to have high correlation with a class and low correlation with any other features. Based on this concept, we measure the correlation between a feature and a class by computing the Information Gain (IG) and the correlation between a pair of features by computing the Mutual Information (MI).

Let $X = [x_1, x_2, ..., x_n]$ denote an n-dimensional feature vector that describes our feature space. Let $S = \{X_1, X_2, ..., X_m\}$ denote the set of training samples for a given user. Each training sample corresponds to a vector of feature values $X_j = [x_{ij}]_{1 \leq i \leq n}$,

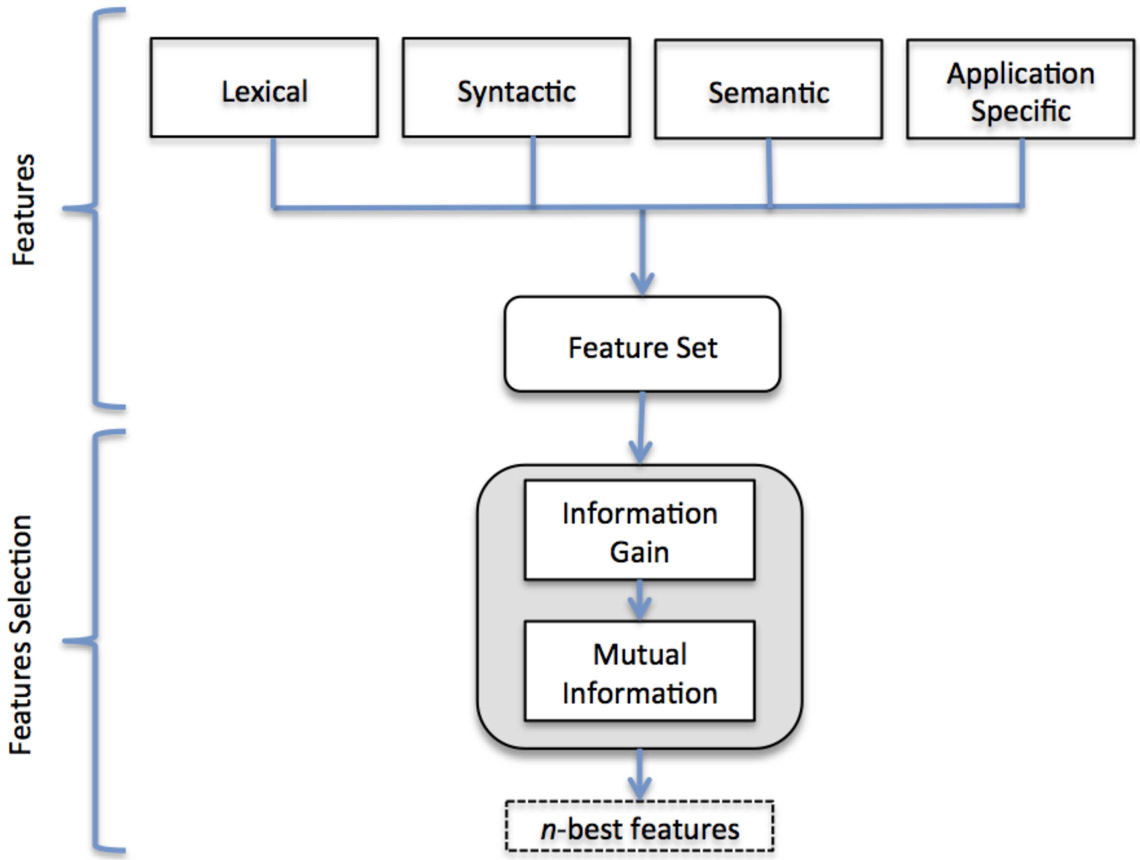


Figure 4.4: Proposed feature selection approach

where x_{ij} is the value of feature x_i for sample X_j.

The information entropy of feature x_i denoted $H(x_i)$ is defined by:

$$H(x_i) = -\sum_{j=1}^{m} p(x_{ij}) log_2 p(x_{ij}) \tag{4.4.1}$$

Where $p(x_{ij})$ denote the probability mass function of x_{ij}.

Given a variable y, with samples $(y_1, ..., y_M)$, the conditional entropy of x_i given y, denoted $H(x_i|y)$, is defined as:

$$H(x_i|y) = -\sum_{j=1}^{m}\sum_{k=1}^{M} p(x_{ij}, y_k) log_2 p(x_{ij}|y_k) \tag{4.4.2}$$

Where $p(x_{ij}, y_k)$ denote the joint probability mass function of x_{ij} and y_k.

Suppose that the dataset is composed by two classes (positive and negative). The IG for a feature x_i with respect to a class is computed as follows:

$$IG(Class, x_i) = H(Class) - H(Class|x_i) \tag{4.4.3}$$

Given two features x_i and x_k, their mutual information (MI) is calculated as follows:

$$MI(x_i, x_k) = H(x_i) - H(x_i|x_k) \tag{4.4.4}$$

For the purpose of feature selection, we retain only features with non-zero information gain and remove a feature when the mutual information is higher then 95%. By computing the IG for features and MI for pairs of features, features with very little or no predictive information and high correlation are identified and removed for each user. In the end, each user ends up with a subset of features that is specific to their individual profile.

4.5 Summary

In this chapter, we present in detail and discuss available stylometric features. Many linguistic features have been suggested for authorship verification, for instance, choice of particular words and syntactic structures [7]. In contrast with topic-based text categorization whose central point is “bags of content words”, stylometric feature spaces suggested in the literature combine lexical, syntactic, semantic and application specific features. Such combination could better expresses the author’s style. We use in our work an initial feature set consisting of existing features, and introduce new n-gram features derived using a supervised learning model.

The chapter also introduces a technique for feature selection. Feature selection for a CA system consists of identifying and keeping only the most discriminating features for each individual user. This allows reducing feature space dimensionality by removing irrelevant attributes and improves the processing time for training and classification.

In the next chapter, we conduct several experiments using shallow classification techniques, assessing their ability for authorship verification.

Chapter 5

Shallow Classifiers

It has been shown that shallow classification architectures can be effective in solving many stylometric analysis problems [68, 94]. A shallow architecture refers to a classifier with only one or two layers responsible for classifying the features into a problem-specific class. Examples of shallow classifiers with one layer include k-Nearest Neighbor (k-NN), Naïve Bayes, Hidden Markov Model (HMM), multi-layer perceptrons (MLPs) with a single hidden layer, and Support Vector Machines (SVM). Examples of shallow classifiers with two layers include SVM-Logistic Regression (SVM-LR), where the output of the SVM is submitted to a logistic function.

In this chapter, we analyze our global feature space using selected shallow classifiers. This shows increased effectiveness of our approach compared to the existing approaches published in the literature, when applied for authorship verification based on short texts. Furthermore, we investigate shorter messages, which are required for continuous authentication systems to operate with reduced window size for re-authentication.

This chapter is organized as follows. In Section 5.1, we give an overview of the shallow classifiers considered in our study, namely, Logistic Regression, SVM, and a hybrid classifier that combines SVM and Logistic Regression. In Section 5.2, we describe our evaluation method. In Section 5.3, we investigate the effectiveness of our stylometric model using the above shallow classifiers. We summarize the chapter in Section 5.4.

5.1 Classifiers Overview

Our n-gram experiment varying f and m showed a slight increase of FRR and FAR across different size of n-grams. However, we note that a block can be classified correctly by one configuration and misclassified by another, suggesting that a combination of different configurations submitted to a machine learning classifier (e.g. SVM, Logistic Regression) could improve the overall classification performance.

In our approach, we decompose an online document into consecutive blocks of short texts over which (continuous) authentication decisions happen. We extract the initial set of features based on the global feature space outlined in the previous chapter, and apply feature selection techniques. In order to balance the dataset, we define a weight for the instances based on the proportion of positive and negative training samples.

In this chapter, we study three different classifiers: logistic regression, SVM, and an hybrid classifier that combines logistic regression and SVM. We give an overview of these classifiers in this subsection, and analyze in subsequently their performances.

5.1.1 Logistic Regression

Logistic Regression (LR) is a well-known and efficient probabilistic statistical classification model [15]. LR is applied for binary classification ($y \in \{0, 1\}$) and Multinomial LR is applied for multi-class problems ($y \in \{0, 1, 2, ...n\}$). The LR prediction is based on the logistic function that is a common sigmoid function. The logistic regression predicts whether a feature vector x belongs to a class y_i by computing the following:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{5.1.1}$$

The output of the logistic function is always between zero and one ($0 \leq f(x) \leq 1$); values close to one indicate high probability that the event will occur, whereas values close to zero indicate the opposite. The graph of a logistic function has a S-curve shape as depicted by Figure 5.1.

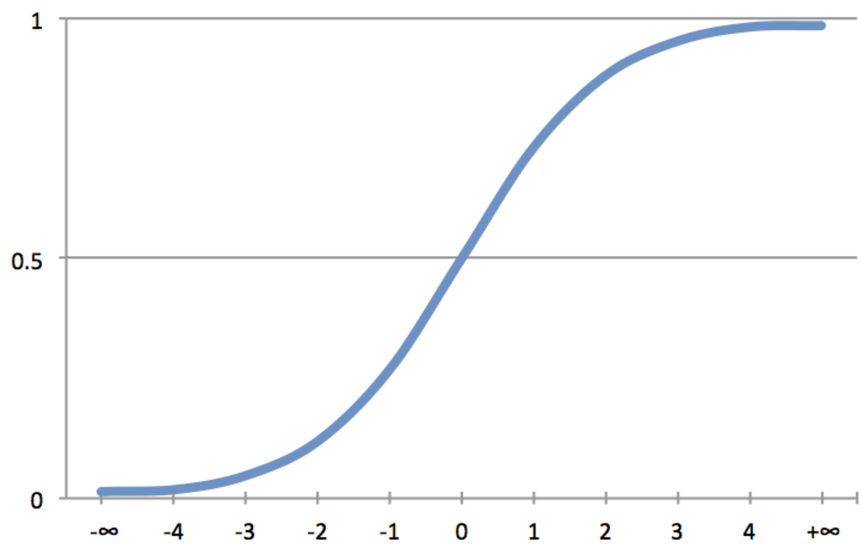


Figure 5.1: The logistic regression curve

5.1.2 SVM

SVM is a binary classifier originally proposed by Vapnik [100]. SVM is based on the idea of mapping the original finite-dimensional space X into a much higher-dimensional space F and building a hyperplane separating points of the two classes (positive and negative). The straight line that divides the two classes are called the optimal hyperplane and the decision boundary is the maximum-margin hyperplane or the largest minimum distance to the training examples, as illustrated in Figure

5.2. The instances that lie closest to the hyperplane are called the support vectors. Training an SVM consists of identifying the support vectors within the training samples.

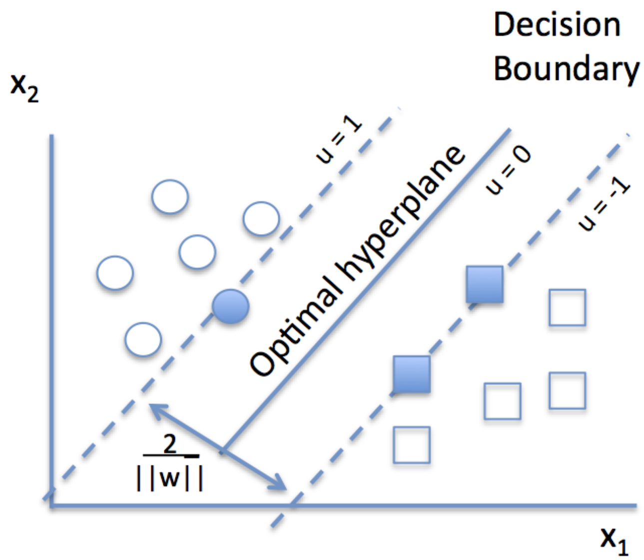


Figure 5.2: Decision boundary separating two classes; samples on the margin are called the support vectors.

Assume that $s_i \in X$ are the support vectors, each associated with a class label $y_j \in \{+1, -1\}$ (for positive and negative examples, respectively). Given an unlabeled sample $x \in X$, classification consists of predicting the corresponding label y_j. This is performed using a decision function as follows.

$$f(x) = \sum_i y_i \alpha_i K(x, s_i) + b \tag{5.1.2}$$

Where α_i are Lagrangian multipliers, and K is a kernel function that measures the similarity or distance between the unlabeled sample x and the support vector s_i. The kernel function $K(x, s_i)$ maps the sample space X into a high-dimensional feature space F. Examples of kernel functions include linear, polynomial, and Gaussian [100]. Table 5.1 shows some examples of kernel functions, and Figure 5.1 shows the effect of different types of kernels for SVM.

Table 5.1: Kernel functions

Kernel type	Inner product kernel
Linear	$K(x,y) = (x \times y)$
Polynomial	$K(x,y) = (x \times y + 1)^p$
Gaussian	$K(x,y) = e^{-\|\|x-y\|\|^2/2\gamma^2}$

5.1.3 SVM-LR

Although SVM is a non-probabilistic classifier, probability estimates can be obtained by integrating SVM with logistic regression into a more robust hybrid classifier (referred to as SVM-LR) [29, 102]. The output of the SVM ($f(x)$) from Equation 5.1.2 is submitted to a logistic function, defined as:

$$P(x) = \frac{1}{1 + e^{-f(x)}} \tag{5.1.3}$$

Where the output of $P(x)$ is always between zero and one ($0 \leq P(x) \leq 1$).

5.2 Evaluation Method

We implemented our framework in Java and utilized the WEKA (Waikato Environment for Knowledge Analysis)[1] machine learning framework and libraries for our classification algorithms [105]. We used a SVM learner called Sequential Minimal Optimization (SMO) and a Logistic Regression learner called SimpleLogistic in WEKA [74, 85].

We evaluated the approach using initially the Enron dataset, and subsequently the Twitter dataset. In order to simulate CA, all messages per author were grouped

[1] available at `http://weka.wikispaces.com`

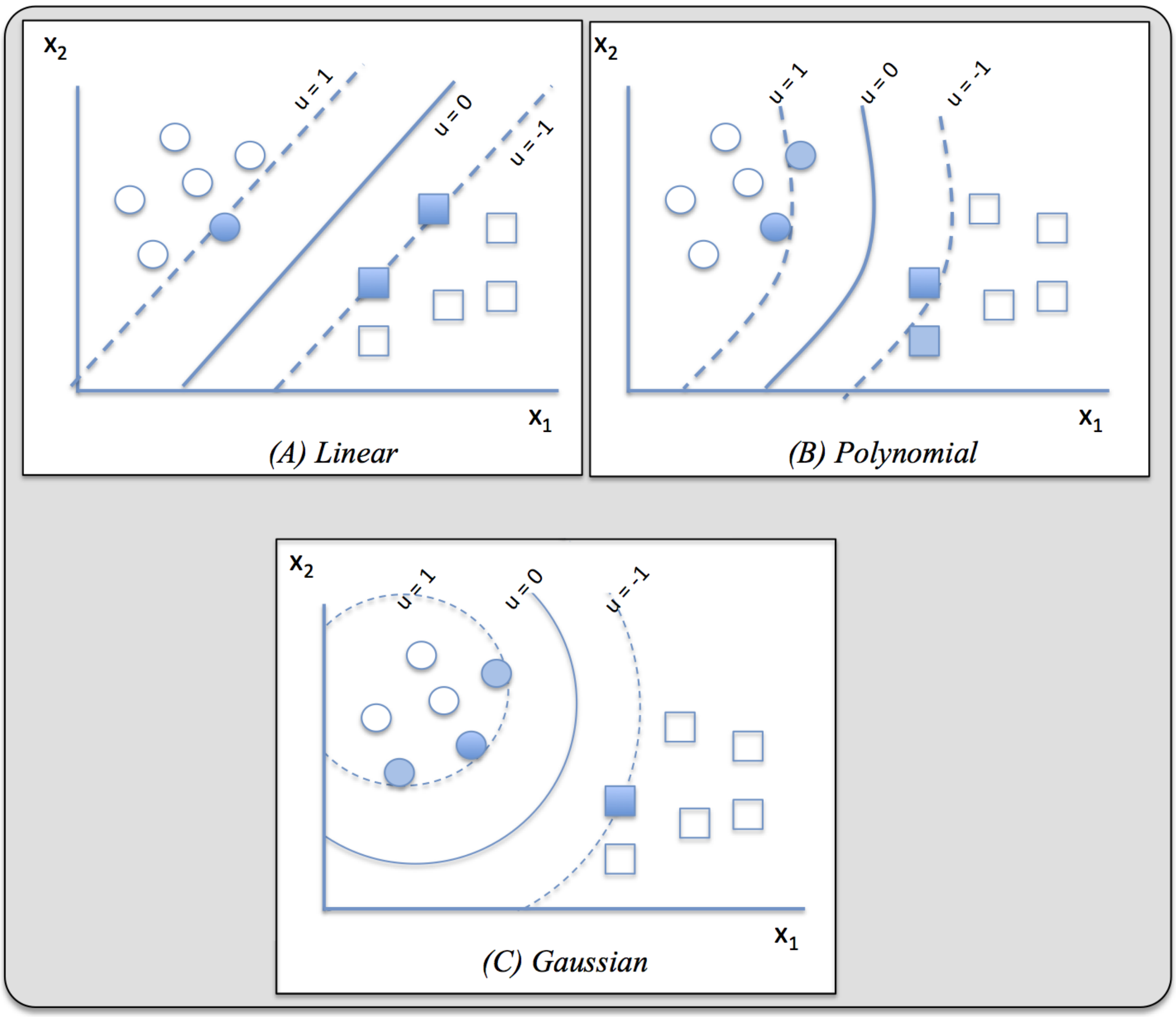


Figure 5.3: The effect of different types of kernels for SVM

creating a long text or stream of characters that was divided into blocks. CA occurs by performing authentication decisions repetitively over consecutive blocks of data captured during a session. We performed our tests with a block size of 140, 280, and 500 characters on average and 50, 100, and 200 blocks per author.

After the preprocessing phase, the Enron dataset was reduced from 150 authors to 76 authors to ensure that only users with 50 instances and 500 characters per instance were involved in our analysis. The number of users in the Twitter dataset remained 100.

In order to evaluate the accuracy of the proposed approach, we performed a 10-fold cross-validation test. We randomly sorted the dataset, and allocated in each (validation) round 90% of the dataset for training and the remaining 10% for testing. For each user U, we computed a corresponding profile by using their training data and training data from other users considered as impostors.

For the Enron dataset, each individual user profile was built using 45 positive instances and 3375 ($= 75 \times 45$) negative instances. The remaining instances consisting of 5 positive instances and 375 ($= 75 \times 5$) negative instances were used for testing. The test was repeated 76 times by considering each time one of the users in our experiment as a legal user while the remaining users were considered as impostors.

For the Twitter dataset, the number of samples used to build the user's profile varies across different block sizes. The profile for the user was built using 45, 90, and 180 positive instances, and 4455, 8910, and 17820 negative instances, when the number of blocks per user were 50, 100, and 200, respectively. As shown in Table 5.2, the testing dataset is composed by the remaining instances, which correspond to 5, 10, and 20 positive instances, and 495, 990, and 1980 for negative instances for block size of 50, 100, and 200, respectively.

We evaluated our method by computing the FRR, FAR, and EER.

Table 5.2: Number of instances used to build the user's profile and perform the evaluation using Twitter dataset

Blocks per user	Training Dataset		Testing Dataset	
	Positive	Negative	Positive	Negative
50	45	4455	5	495
100	90	8910	10	990
200	180	17820	20	1980

5.3 Evaluation Results

We started our evaluation by conducting baseline experiments, through which some of our model parameters were tuned. For these baseline experiments, we used the Enron dataset and analysed block of texts with 500 characters. Following the baseline experiments, we conducted some experiments to compare the performance of the selected shallow classifiers (i.e. SVM, SVM-LR and LR). A third set of experiments were then conducted that focused on reducing the block size using the Twitter dataset. Finally, a set of experiments were carried out comparing the processing speed of SVM and LR.

5.3.1 Baseline Experiments

We describe in this section different experiments undertaken to study the impact of class imbalance, feature selection approach, and choice of SVM kernel.

These experiments were conducted using the Enron dataset involving 76 authors with a block size of 500[2] characters[3] on average and 50 blocks or instances per user, since this configuration yielded the best performance results in the previous experiments (see Section 4.2).

[2]The block size starts after punctuation and ends with an entire word, which means if 500 characters corresponds exactly to the middle of a word, then the block will have 500 characters plus the rest of the characters involved in the complete word.

[3]include A..Z, a..z, 0..9, punctuation, white space, special characters

Balancing Class Distribution

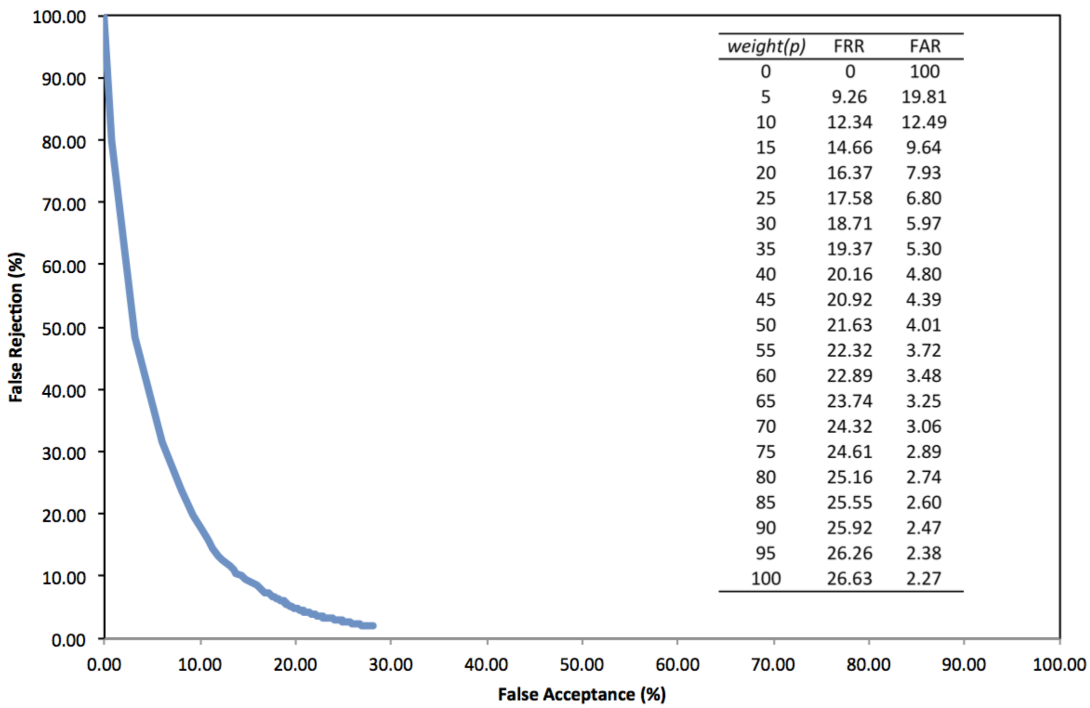
Our classification model consists of two classes. The first class is composed by (positive) samples from the author, whereas the second class is composed by (negative) samples from other authors. Thereby, the negative class has more samples than the positive class, generating imbalance class distribution. Our approach to deal with this situation is to assign a weight P (denoted $weight(P)$) to the negative class corresponding to the ratio between the total number of positive samples and the total number of negative samples.

In order to test the effect of the $weight(P)$, an experiment was conducted using the SVM linear kernel and information gain as feature selection technique. Figure 5.4 shows the receiver operating characteristic curve for the experiment. The curve shows the relation between the FAR and FRR when varying $weight(P)$ from 0 to 100. The optimal performance achieved by the system was obtained when setting the $weight(P)$ limit to 10, with $FAR = 12.49\%$, $FRR = 12.34\%$, and $EER = 12.42\%$. Likewise, subsequent experiments used $weight(P)$ set to 10.

Feature Selection Technique and Parameters

In this experiment, we explored different threshold values for Information Gain and performed tests using Mutual Information. We used SVM classifier with linear kernel and set the weight P value to 10. Different tests were performed by setting the information gain to be greater than 0, 0.005 and 0.010, yielding EER of 12.42%, 14.54% and 15.27%, respectively.

Next, we extended the (information gain) feature selection technique by adding the mutual information selection approach and setting the information gain to be greater than 0. This yields EER of 12.05%, as shown in figure 5.5. When the feature selection was omitted, we obtained as performance an EER of 12.08%. This indicates that our



weight(p)	FRR	FAR
0	0	100
5	9.26	19.81
10	12.34	12.49
15	14.66	9.64
20	16.37	7.93
25	17.58	6.80
30	18.71	5.97
35	19.37	5.30
40	20.16	4.80
45	20.92	4.39
50	21.63	4.01
55	22.32	3.72
60	22.89	3.48
65	23.74	3.25
70	24.32	3.06
75	24.61	2.89
80	25.16	2.74
85	25.55	2.60
90	25.92	2.47
95	26.26	2.38
100	26.63	2.27

Figure 5.4: Receiver Operating Characteristic curve obtained by varying $weight(P)$ using SVM as classifier and the Enron dataset involving 76 authors with block size of 500 characters and 50 blocks per user. The results were obtained using a linear kernel for SVM and Information Gain as feature selection technique.

feature selection technique has negligible impact on the accuracy of the classifier. However, feature selection is still beneficial in terms of reduction in processing time due to the reduction in the number of features. On average, the feature set is reduced from 972 features to 242 when feature selection takes place compared to when it is omitted.

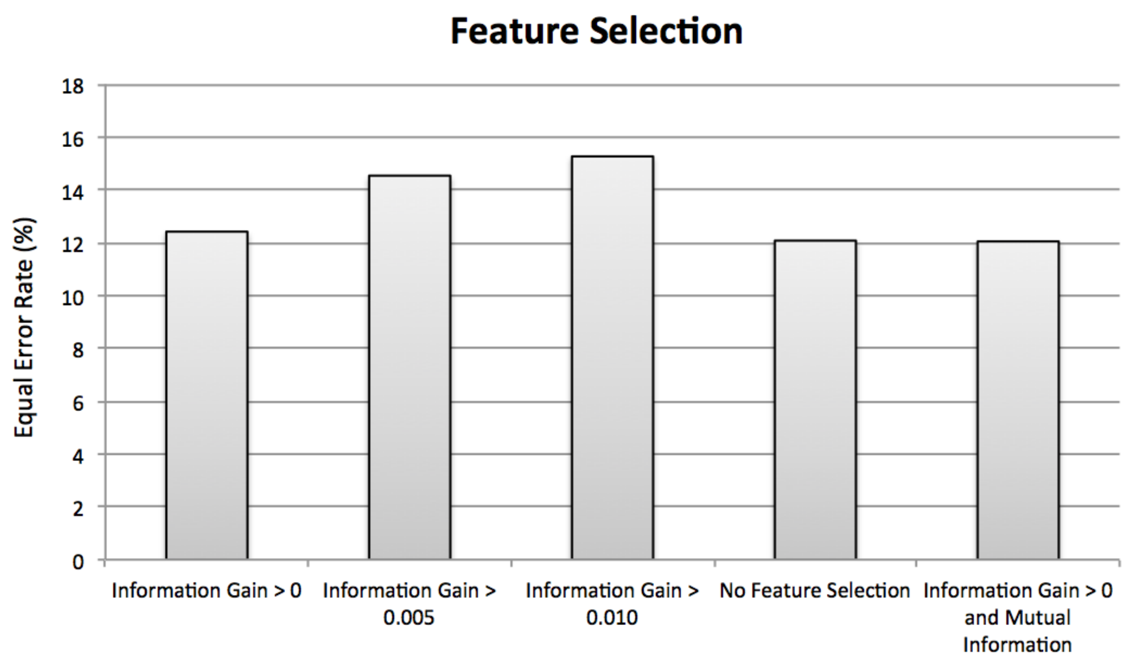


Figure 5.5: Experiments comparing the impact of the feature selection method

Varying the SVM Kernel

We performed a set of experiments to determine what is the best kernel for our research. The first experiment used a linear kernel, yielding EER of 12.05%. Subsequent experiments using polynomial kernels degree 3 and degree 5, and also Gaussian kernel yielded EER varying from 15.27% to 18.95%, as shown in Table 5.3. From the above results, it can be concluded that the hyperplane separating positive from negative data is linear. Therefore, the subsequent experiments were run with linear kernel only.

Table 5.3: EER obtained by varying the type of SVM Kernels

SVM Kernel	**EER %**
Linear	12.05
Polynomial 3	15.28
Polynomial 5	18.95
Gaussian	18.07

Comparison among different SVM kernels based on the Enron dataset involving 76 authors with block size of 500 characters and 50 blocks per user.

5.3.2 Comparison with Different Classifiers

Previous experiments focused on exploring different strategy to balance the dataset and to test the effect of different kernels for SVM. This allowed us to define the best configuration for SVM and also have a baseline. Considering the baseline, we conducted further experiments to study and compare the performance of the other shallow classifiers considered in our work.

These experiments were also based on the Enron dataset involving 76 authors, block size of 500 characters, and 50 blocks per author. Feature selection was performed using Information Gain and Mutual Information approaches. Table 5.4 shows the performance for SVM, SVM-LR and LR classifiers, where EER of 12.05%, 9.98% and 9.18% were obtained, respectively. These results show that SVM-LR and LR perform much better than SVM, while LR outperforms the 2 other classifiers.

Table 5.4: Authorship verification using the Enron dataset

	SVM	**SVM-LR**	**LR**
EER (%)	12.05	9.98	9.18

Authorship Verification with 76 authors, block size of 500 characters, and 50 blocks per author. Feature selection was performed using the Information Gain and Mutual Information approaches.

5.3.3 Analysing Short Messages

An important aspect of continuous authentication systems is to re-authenticate the user in a small window of time. In stylometry-based authorship verification for continuous authentication, shorter authentication delay corresponds to analyzing smaller data samples. In order to achieve this goal, we explore smaller blocks of texts by analysing micro messages from the Twitter dataset while trying to maintain acceptable accuracy.

An initial experiment was carried out involving 100 authors with block size of 140 characters and 100 blocks per user, as illustrated in table 5.5. These tests were performed with SVM, SVM-LR and LR classifiers, and used the information gain and mutual information selection approaches. EER of 23.49%, 21.45% and 19.05% were obtained when using SVM, hybrid SVM-LR and LR classifiers, respectively.

Increasing the training set and block size affect the accuracy. For instance, when increasing the number of blocks per user to 200, we obtained EER of 20.27%, 18.37% and 16.74%, for SVM, SVM-LR and LR classifiers, respectively. Also, using a block size of 280 characters and 50 blocks per user, our results reached EER of 18.47%, 17.83% and 16.16% for hybrid SVM-LR and LR classifiers, respectively. Using 100 blocks per user, we obtained EER of 14.87%, 13.27% and 11.83% for SVM, SVM-LR and LR classifiers, respectively.

5.3.4 Classification Speed

We also examined the classification speed for SVM-LR and LR. Table 5.6 illustrates the processing time measured in seconds for different experiments and classifiers. The column "Train" shows the processing time required to train the profile of a single author. The performance is computed by varying the number of features and training samples. In fact, the number of features has the most significant impact on the

Table 5.5: Authorship verification using the Twitter dataset

Block Size	Blocks per user	**SVM %**	**SVM-LR %**	**LR %**
140	100	23.49	21.45	19.05
	200	20.27	18.37	16.74
280	50	18.47	17.83	16.16
	100	14.87	13.27	11.83

EER for SVM, SVM-LR and LR using Twitter dataset involving 100 authors and varying the size of the block and the number of blocks per author. Feature selection was performed by Information Gain and Mutual Information approaches.

performance. For instance, using SVM-LR classifier, the required time to train a single user with 3,420 training samples and 242 features was 1.30 seconds. On the other hand when the number of features decrease to 147 and the number of training samples increases to 4,500, the overall time decreases to 1.16 seconds. Furthermore, results demonstrated that LR requires substantially more processing time to train a classifier than SVM-LR; on average LR is 22 times slower than SVM-LR. All experimental tests were performed on a Dell C6100 computer with twelve 2.66-GHz Xeon x5650 cores and 24 GB of RAM. The experiments were run in a serial job and used only one core at a time.

5.4 Summary

In this chapter, we investigate the possibility of using stylometry for authorship verification for short online messages using shallow classifiers. The approach taken is to start with large chunks of text to simulate continuous authentication and decrease the size of chunk in order to simulate small windows authentication. Block sizes of 500, 280, and 140 characters are investigated. The problem is addressed as a two-class classification problem composed by positive and negative samples.

Comprehensive experiments based on 2 different datasets demonstrate that the

Table 5.6: Processing time for the different classifiers

Dataset	Number of blocks / Block size	Training Samples	Test samples	Features	SVM-LR*		LR*	
					Train	1 Fold	Train	1 Fold
Enron (76)	50/500	3420	380	242	1.30	304.09	28.25	2632.21
Twitter (100)	50/140	4500	500	147	1.16	324.33	12.07	2577.38
	50/280	4500	500	290	1.49	578.10	25.32	4833.40
	100/140	9000	1000	211	3.10	1161.92	51.14	8340.87
	100/280	9000	1000	452	4.65	1837.21	149.61	14533.69
	200/140	18000	2000	220	5.36	3064.92	114.69	21389.95

* Time unit is expressed in seconds.

proposed approaches achieve promising results when compared to existing work in the literature. Although the results are very promising, the proposed approaches using shallow classifiers still face significant challenge when the size of text for analysis decreases. We concluded that our results could be improved by expanding the feature set and using more powerful classifiers. These considerations will be explored in the subsequent chapters.

Chapter 6

Feature Merging

Previous chapters demonstrated the possibility to use stylometry for continuous authentication. We proposed different types of features and used shallow classifiers to discriminate between different authors. Although the obtained evaluation results are promising, there is a need to improve these results for CA. Investigating new features is one obvious way that could improve the results. In this chapter, we propose an approach to compute new features by merging existing ones. We assess the performance of the proposed feature generation approach by performing a series of experiments using the Enron and Twitter datasets.

This chapter is structured as follows. We describe our feature merging approach and present the revised global feature set in section 6.1. Section 6.2 discusses our classification method. Section 6.3 presents the evaluation method. Section 6.4 outlines the experimental results. Finally, we summarize the chapter in Section 6.5.

6.1 Features Merging Approach

Zhou and colleagues achieved significant performance improvement in generative tasks (e.g., minimizing reconstruction error) and discriminative tasks (e.g., minimizing supervised loss function) by merging similar features [109]. They used cosine distance to find the most similar feature pairs for merging, and applied linear combination to generate the new features.

We propose a new method to merge a pair of features into a single feature that considers only the information gain as selection criteria.

Let $X = [x_1, x_2, ..., x_n]$ denote an n-dimensional feature vector that describes our feature space. Let $S = \{X_1, X_2, ..., X_m\}$ denote the set of training samples for a given user. Each training sample corresponds to a vector of feature values $X_j = [x_{ij}]_{1 \leq i \leq n}$, where x_{ij} is the value of feature x_i for sample X_j.

The information entropy of feature x_i denoted $H(x_i)$ is defined by:

$$H(x_i) = -\sum_{j=1}^{m} p(x_{ij}) log_2 p(x_{ij}) \tag{6.1.1}$$

Where $p(x_{ij})$ denote the probability mass function of x_{ij}.

Given a variable y, with samples $(y_1, ..., y_M)$, the conditional entropy of x_i given y, denoted $H(x_i|y)$, is defined as:

$$H(x_i|y) = -\sum_{j=1}^{m} \sum_{k=1}^{M} p(x_{ij}, y_k) log_2 p(x_{ij}|y_k) \tag{6.1.2}$$

Suppose that the dataset is composed by two classes (positive and negative). The IG for a feature x_i with respect to a class is computed as follows:

$$IG(Class, x_i) = H(Class) - H(Class|x_i) \tag{6.1.3}$$

Given two features x and y, let $P_y(x)$ denote the following ratio:

$$P_y(x) = \frac{IG(x)}{IG(x) + IG(y)} \tag{6.1.4}$$

Let x_i and x_k denote two features to be merged in a new feature x_r. The merging consists of computing the values of features x_r from the training samples; the merged values are computed as follows:

$$x_{rj} = P_{x_k}(x_i) \times x_{ij} + P_{x_i}(x_k) \times x_{kj} \tag{6.1.5}$$

The decision to keep the new feature is made by comparing the corresponding information gain $IG(x_r)$ to $IG(x_i)$ and $IG(x_j)$, respectively. The new feature x_r is added to the feature set if and only if $Max(IG(x_i), IG(x_k)) < IG(x_r)$. In this case feature x_r is added to the feature set while features x_i and x_k are removed from the set. The above process is repeated for all features by comparing two features at a time.

Since some features have different ranges of values, we pre-process the selected features before merging them. The pre-processing consists of normalizing the feature values between 0 and 1, and discretizing the numeric feature values into binary values (0 and 1) using Fayyad and Irani discretization approach [38, 70]. The new features created after completing the merging process are also normalized between 0 and 1 and then added to the features list.

6.1.1 Updated Feature Set

Our updated global feature set includes the features identified in the previous chapters, the new features derived using our merging technique, and the 50 most frequent 2-grams and 3-grams words per author. Our new global feature set consists of 528 lexical character features, 175 lexical word features, 362 syntactic features, 7 application specific features, and the merged features, whose number vary from one author to another. The new global feature space is depicted in Table 6.1.

Table 6.1: List of the updated stylometry features used in this chapter

Feature	**Characteristics**
<u>Lexical</u>	
F_1 .. F_5	Number of characters (C), lower character/C, upper characters/C, white-space/C, vowels (V)/C
F_6 ...F_{10}	Vowels (a, e, i, o, u) / V
F_{11} ...F_{36}	Alphabets (A-Z) / C
F_{37}	Number of special characters (S) /C
F_{38} ...F_{50}	Special Characters (e.g. '@', '#', '$', '%', etc.) / S
F_{51} ...F_{67}	Character 5 and 6-grams ($r_U(b)$)
F_{68} ... F_{192}	Text based icon (8 groups)
F_{193} ...F_{272}	Unicode - emoticons (code range from 1F600 to 1F64F)
F_{273} ...F_{528}	Unicode - miscellaneous symbols (code range from 2600 to 26FF)
F_{529}	Total number of words (N)
F_{530} ...F_{539}	Average sentence length in terms of words /N
F_{540}	Number of words longer than 6 characters/N
F_{541}	Total number of short words (1-3 characters)/N
F_{542}... F_{543}	Average word length and syllable per word
F_{544}	Ratio of number of characters in words to N
F_{545} ...F_{550}	Number of replaced words / N
F_{551} ...F_{600}	The 50 most frequent words per author
F_{601} ...F_{650}	The 50 most frequent *2*-grams words per author
F_{651} ...F_{700}	The 50 most frequent *3*-grams words per author
F_{701}... F_{702}	*Hapax legomena* and *dis legomena*
F_{703}	Vocabulary richness (total different words/N)
<u>Syntactic</u>	
F_{704}	Total number of punctuation (P)
F_{705} ...F_{712}	Punctuation divided by P
F_{713} ...F_{824}	General punctuation (code range from 2000 to 206F)
F_{825} ...F_{829}	Total number of conjunction, interrogative, preposition, interjection, and pronouns each one divide by N
F_{830} ...F_{1065}	Ratio of functional word divide by the respective total word group
<u>Application-specific</u>	
F_{1066} ... F_{1067}	Total number of sentences and number of paragraphs
F_{1068} ...F_{1070}	Average of characters, words and sentences in a block
F_{1071} ... F_{1072}	Number of sentences beginning with upper and lower case
<u>Merging features</u>	
F_{1073} ...	Merging features vary from one author to another

6.2 Classification

We follow the same approach described in previous chapters by decomposing an online document into consecutive blocks of short texts over which (continuous) authentication decisions happen. We extract the set of features based on the global feature space outlined in Section 6.1.1, and apply feature selection technique. In order to balance the dataset, we use the weighting approach defined in the previous chapter.

In this chapter, we study three shallow classifiers: SVM, SVM-LR, and logistic regression.

6.3 Evaluation Method

We implemented our framework in Java and utilized a SVM learner called Sequential Minimal Optimization and a Logistic Regression learner called SimpleLogistic in WEKA [74, 85].

We evaluated the approach using initially the Enron dataset, and subsequently the Twitter dataset. We address the problem as a two-class problem composed by (positive) samples from the author and by (negative) samples from other authors. All messages per author were grouped creating a long text or stream of characters that was divided into blocks. We performed our tests with a block size of 140, 280, and 500 characters on average and 50, 100, and 200 blocks per author. We set the value of the weight P to 10. Feature selection was carried out by using IG and MI and by setting the IG to be greater than 0.

After the preprocessing phase, the Enron dataset was reduced from 150 authors to 76 authors in order to ensure that only users with 50 instances and 500 characters per instance were involved in our analysis. The number of users in the Twitter dataset remained 100.

We performed a 10-fold cross-validation test and randomly sorted the dataset, and allocated in each (validation) round 90% of the dataset for training and the remaining 10% for testing. For each user U, we computed a corresponding profile by using their training data and training data from other users considered as impostors.

We evaluated our method by computing the FRR and FAR, where the overall FRR and FAR were obtained by averaging the individual measures over the entire user population. The EER was determined by identifying the operating point where FRR and FAR have the same value.

6.4 Evaluation Results

Initial experiments were performed to determine the effects of the feature merging using the Enron dataset. In the next set of experiments, we compared the performance of the selected shallow classifiers (i.e. SVM, SVM-LR and LR). Finally, we performed a set of experiments using the Twitter dataset.

6.4.1 Baseline Experiments

Our baseline experiments were conducted by using the Enron dataset involving 76 authors with a block size of 500 characters on average and 50 blocks per user. Our first experiment used the default feature set described in Section 6.1.1 (so without the new word n-grams mentioned above), applied feature merging, and used SVM with linear kernel as a classifier. The experiment yielded an EER of 11.48%. This corresponds to a reduction of the error rate by 4.7%, when compared with similar experiments without feature merging presented in Chapter 5 (see Table 5.4). Likewise, feature merging has a positive impact on the accuracy of the system.

In order to test the effect of the (new) word n-gram feature, we used the updated global feature set (introduced in Section 6.1.1), while keeping the rest of the configuration the same. The experiment yielded EER of 11.09%, as shown in Table 6.2. This indicates a modest increase in performance when adding adding the (new) word n-gram to our global feature set. Subsequent experiments used the default feature set with the feature merging, since the approach yielded (slightly) improved performance results.

Table 6.2: Baseline experiments using the Enron dataset

	Merging without word n-gram	**Merging with word n-gram**
EER (%)	11.48	11.09

Experiments using SVM with linear kernel on Enron dataset involving 76 authors with block size of 500 characters and 50 blocks per user. The results show the EER using the feature merging approach and the word n-gram approach.

6.4.2 Email dataset

Building on the results obtained in the baseline experiments, we conducted further experiments to evaluate the performance of our feature merging approach with different shallow classifiers. The experiments were based on the Enron dataset involving 76 authors with a block size of 500 characters on average and 50 blocks per user. The performances obtained when using feature merging and word n-gram are summarized in Table 6.3 for SVM, SVM-LR and LR classifiers. The obtained results confirm the trend observed in the previous experiment where adding word n-gram to the global feature set achieves modest improvement in performance for the different classifiers, with LR still outperforming the other classifiers.

Table 6.3: Experiments using shallow classifiers on the Enron dataset

Approach	**SVM (%)**	**SVM-LR (%)**	**LR (%)**
Merging without word n-gram	11.48	9.56	8.89
Merging with word n-gram	11.09	9.35	8.72

EER for SVM, SVM-LR, and LR on Enron dataset involving 76 authors with block size of 500 characters and 50 blocks per user.

6.4.3 Twitter Dataset

In this section, we evaluate our feature merging approach by decreasing the block size. We performed a series of experiments on the Twitter dataset using block size of 140 and 280 characters with 50, 100, and 200 blocks per user.

Table 6.4 shows detailed experimental results for short messages. When the number of blocks per user is 100 and the block size is 140 characters, the EER is 18.95%, 17.51%, and 17.43% for SVM, SVM-LR, and LR, respectively. When the character block size increases to 280, the EER drops to 12.34%, 10.55%, and 10.27% for SVM, SVM-LR, and LR, respectively. On the other hand, if we keep the block size to 280 characters and drop the number of blocks per user to 50, EER increases to 16.99%, 15.98%, and 15.20% for SVM, SVM-LR, and LR, respectively. These results show that the block size and the number of blocks per user affect directly the results. In addition, the obtained results with a relatively short block of 280 characters show an improvement of the classification accuracy when compared with the literature. However, the EER is still high for continuous authentication. Therefore, there is a need to investigate other classifiers in order to improve the accuracy of the proposed authentication system.

Table 6.4: Experiments using shallow classifiers on the Twitter dataset

Block Size	Blocks per user	**SVM (%)**	**SVM-LR (%)**	**LR (%)**
140	100	18.95	17.51	17.43
	200	17.25	16.70	16.33
280	50	16.99	15.98	15.20
	100	12.34	10.55	10.27

EER for SVM, SVM-LR, and LR using Twitter dataset involving 100 authors and varying the block size of characters and the number of blocks per author.

6.5 Summary

In this chapter, we expanded our feature set by introducing a technique for generating new features through merging. We evaluated our approach by conducting a series of experiments using the Enron and Twitter datasets. Although the performance results are very promising, there is a need to explore further improvements. This can be done by investigating more powerful classifiers.

In this regard, we investigate in the next chapter deep learning models, which have emerged as more effective alternative to shallow machine learning techniques for certain class of problems.

Chapter 7

Deep Learning Classifier

In this chapter, we use a deep learning classifier and assess the robustness of our proposed approach against forgeries attempts. We investigate the use of deep models for authorship verification, specifically we study Deep Belief Network (DBN). DBN has been shown to be powerful analysis techniques in handwriting recognition, visual detection of objects, and speech recognition, exhibiting an effective encoding learning of a complex distribution in an unsupervised manner [16,34,48,88] . DBN is a type of deep neural network composed of multiple layers of Restricted Boltzmann Machines (RBMs) with a softmax layer added to the top for recognition tasks.

This chapter is structured as follows. Section 7.1 explores and introduces the Deep Belief Network classifier. Section 7.2 details the experimental evaluation. Section 7.3 presents the evaluation results. Finally, we summarize the chapter in Section 7.4.

7.1 Classification

As mentioned earlier, we approach authorship verification as a classification task composed by two-classes. The first class is composed by (positive) samples from the author, whereas the second class is composed by (negative) samples from other authors. In the training phase, we generate a profile for individual users given a feature set and a training set of positive and negative blocks of short texts. We balance the dataset by over-sampling the minority-class [11], in this case the positive samples. To authenticate a user, we match the monitored block of text against the profile for the claimed identity and compute the individual metrics.

In this chapter, we use for classification a generative model consisting of multiple stacked levels of neural network named Gaussian-Bernoulli Deep Belief Network (Gaussian-Bernoulli DBN). The structure of the model is composed by one layer of Gaussian-Bernoulli Restricted Boltzmann Machine, followed by a stack of Restricted Boltzmann Machines, and a top layer with a shallow classifier.

7.1.1 Restricted Boltzmann Machines (RBM)

RBM is a generative stochastic network that learns probability distribution over its set of inputs. RBM is composed by a layer of n visible (*input*) neurons $v = [v_1, v_2, ..., v_n]$ and a layer of m_1 hidden neurons $h = [h_1, h_2, ..., h_{m_1}]$, as illustrated in Figure 7.1. In contrast with the original Boltzmann Machine [92] that allows connection among all units, the Restricted version of a Boltzmann Machine allows connection between visible and hidden units only [47]; there is no connection between units from the same layer.

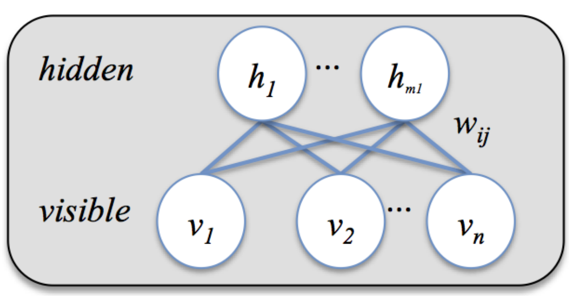


Figure 7.1: Restricted Boltzmann Machine structure composed by visible and hidden units. Each connection is between visible and hidden units only; there is no connection between units from the same layer.

Visible and hidden neurons map multiple signals into one output. The states of hidden and visible neurons are defined as follows:

$$p(h_j = 1|v) = S\left(c_j + \sum_{i=1}^{n} w_{i,j} v_i\right) \tag{7.1.1}$$

$$p(v_i = 1|h) = S\left(b_i + \sum_{j=1}^{m_1} w_{i,j} h_j\right) \tag{7.1.2}$$

Where v_i and h_j are the binary states of visible unit i and hidden unit j, b_i and c_j are the bias vectors on the visible and hidden units, $w_{i,j}$ is the weight between them, and S denotes the sigmoid activation function:

$$S(x) = \frac{1}{1 + e^{-x}} \tag{7.1.3}$$

Standard (Bernoulli-Bernoulli) RBM has binary-valued stochastic neurons in the visible and hidden units, and a joint configuration (v, h) is defined in terms of an energy function *E(v,h)* [45], defined as:

$$E(v, h) = -\sum_{i \in visible} b_i v_i - \sum_{j \in hidden} c_j h_j - \sum_{i,j} -v_i h_j w_{i,j} \tag{7.1.4}$$

The probability distribution over visible and hidden units is given by:

$$p(v, h) = \frac{1}{Z} e^{-E(v,h)} \tag{7.1.5}$$

Where Z is a normalization constant computed by summing over all possible pairs of visible and hidden vectors:

$$Z = \sum_{v,h} e^{-E(v,h)} \tag{7.1.6}$$

The marginal probability that the model assigns to a visible vector v is given by:

$$p(v) = \frac{1}{Z}\sum_{h} e^{-E(v,h)} \tag{7.1.7}$$

Training an RBM consists of minimizing the energy of the network by updating weights and biases. An efficient training algorithm named Contrastive Divergence was proposed by Hinton [46]. The training consists of alternatively sampling the hidden units given visible units $p(h|v)$ and the visible units given hidden units $p(v|h)$. In the Contrastive Divergence algorithm, weights and biases can be updated after a single iteration of Gibbs sampling, as follows:

$$w_{ij} = w_{ij} - \alpha\left(\left\langle v'_i h'_j\right\rangle - \left\langle v''_i h''_j\right\rangle\right) \tag{7.1.8}$$

$$b_i = b_i - \alpha\left(\left\langle v'_i\right\rangle - \left\langle v''_i\right\rangle\right) \tag{7.1.9}$$

$$c_j = c_j - \alpha\left(\left\langle h'_j\right\rangle - \left\langle h'_j\right\rangle\right) \tag{7.1.10}$$

Where α is the learning rate, v' is a training sample, h' is sampled from $p(h|v')$, v'' is sampled from $p(v|h')$, and h'' is sampled from $p(h|v'')$. The angle brackets denotes the expectation over the data distribution. A complete cycle of learning also called "epoch" can be repeated several times.

7.1.2 Gaussian-Bernoulli Restricted Boltzmann Machines

Gaussian-Bernoulli Restricted Boltzmann Machines allow modelling real-valued data in RBM by transforming the data into binary values using Gaussian units in the visible layer [30, 45]. Gaussian-Bernoulli RBM has real values in its visible layer and binary values in its hidden layer. The energy function for a Gaussian-Bernoulli RBM is as follows:

$$E(v,h) = -\sum_{i\in visible} \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{j\in hidden} c_j h_j - \sum_{i,j} \frac{v_i}{\sigma_i} h_j w_{i,j} \tag{7.1.11}$$

Where σ_i is the standard deviation for visible unit i.

The conditional probability of a visible neuron is defined as follows:

$$p(v_i|h) = \mathcal{N}\left(b_i + \sum_{j=1}^{m1} w_{i,j} h_j, 1\right) \tag{7.1.12}$$

Where v_i takes real values and $\mathcal{N}$ is the probability density for normal distribution with mean $\sum_{j=1}^{m1} w_{i,j} h_j + b_i$ and variance one.

7.1.3 Gaussian-Bernoulli Deep Belief Network

Gaussian-Bernoulli Deep Belief Network is a probabilistic generative model that is composed of a single layer of Gaussian-Bernoulli RBM and multiple layers of RBMs followed by a softmax layer [13, 48], as shown in Figure 7.2. The training is semi-supervised performed in two phases, consisting of a *pre-training* phase and a *fine-tuning* phase.

The pre-training phase uses unsupervised learning and is performed incrementally layer-by-layer. Likewise, the first layer of the Gaussian-Bernoulli RBM receives real-valued input. The layer is trained for several epochs. The activation probabilities from the hidden units are then used as the visible data input for the layer up (RBM_1). The same process is repeated for the next layers propagating upward the transformed data.

A softmax is added on top of the last RBM layer. The input of the softmax is the output of the last hidden layer $h^{(l)}$. Fine-tuning is carried through a supervised training phase where the weights are adjusted considering the *inputs* and desired outputs based on the labeled training data. Fine-tuning is performed via supervised gradient descent of the negative log-likelihood cost function.

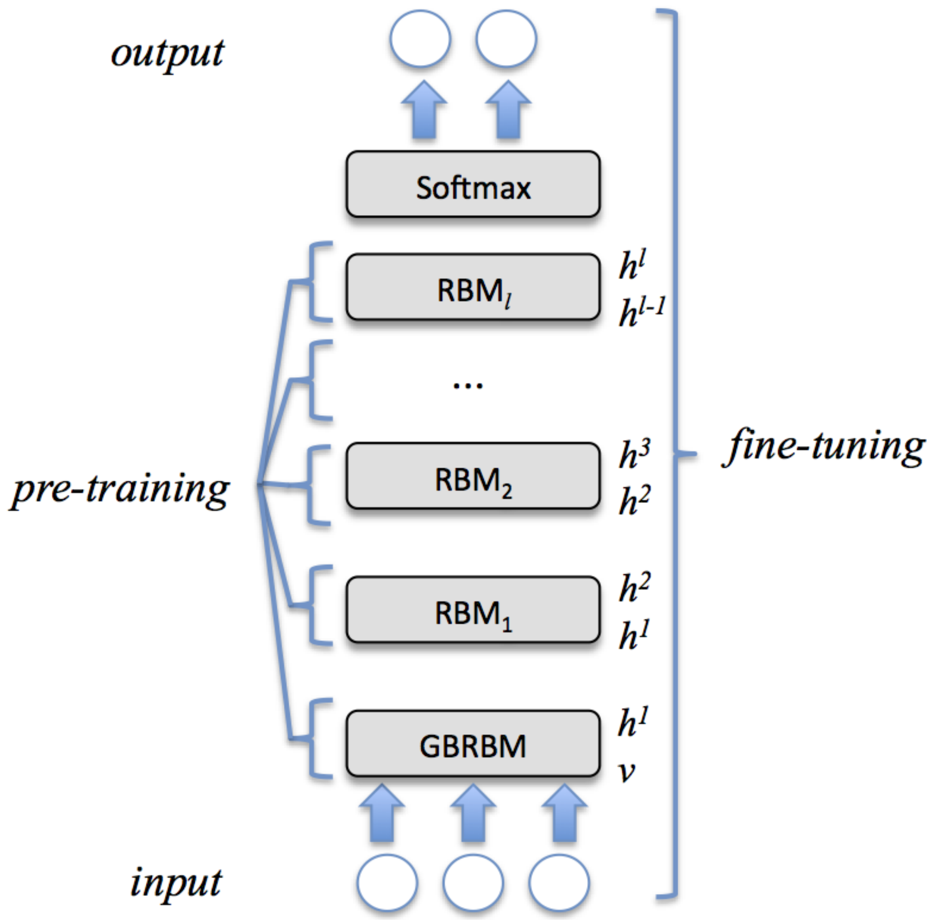


Figure 7.2: Gaussian-Bernoulli Deep Belief Network structure composed by one layer of Gaussian-Bernoulli RBM, l layers of RBMs, and on top of the last layer (RBM_l) is a softmax layer.

7.1.4 Model Settings and Implementation

We used a Java program to perform feature extraction, feature selection, and data preprocessing. We implemented our Gaussian-Bernoulli DBN classifier in python with Theano (on GPU) [14] by adapting the original DBN source code from `http:`

`//deeplearning.net/tutorial`.

Our classifier involved three hidden layers consisting of one Gaussian-Bernoulli RBM layer and two Bernoulli-Bernoulli RBM layers. The softmax was implemented using a Logistic Regression classifier. The input layer consists of n real-valued features varying from 0 to 1, as required by the Gaussian-Bernoulli RBM. The inputs of the layer above are binary values, as defined in the Bernoulli-Bernoulli RBM. The number of hidden units varies from one author to another, since the number of features varies from one author to another. We use three hidden layers in our experiments. In experiments conducted by Sarikaya et al. [91], the use of three hidden layers was found to be more effective than the other hidden layer sizes that they tried. We adopted a linear shape for our network, where the number of hidden units per layer decreases when the number of layers increases. Therefore, after trying different layer decompositions (using sample data), we used for the hidden units in the first, second, and third layers, 75%, 50%, and 25% of the initial feature space, respectively. The other parameters consist of 100 pre-training epoch[1], mini-batches size equal to the number of self samples (45, 90 or 180), unsupervised learning rate of 0.001, and supervised learning rate of 0.01.

In our model, the number of epochs in the fine-tuning phase is variable, since the number of features ($input(x)$) varies from one author to another. Therefore, one author may need fewer epochs to model the training data while another author may need more epochs. In order to avoid over-fitting, our approach to define the appropriate number of epochs is described as follows:

1. Define a variable ve for the validation error and set ve to a desired target value. The validation error corresponds to the percentage of incorrectly classified training samples;

[1] An epoch is a complete learning cycle.

2. The initial epoch e is set to 50;

3. Calculate the current validation error, after performing e epochs;

4. If the current validation error is higher than ve then e is incremented by 50 epochs;

5. Perform steps 3 and 4 until the current validation error is lower than ve;

6. Stop the fine-tuning if the current validation error is lower than 2% or e is higher than or equal 1,000 epochs;

7. Calculate the metrics for the testing dataset when the fine-tuning phase stops.

7.2 Evaluation Method

Authentication consists of computing the similarity of a sample against the profile (corresponding to the claimed identity), and comparing the obtained score S against some threshold Th. If the score is greater or equal to the threshold, the sample will be accepted and considered as genuine. Otherwise, it will be rejected and classified as being from an impostor.

During the above classification process, samples may be wrongly accepted (as genuine) or rejected (as from an impostor). In this context, the accuracy of biometric systems is evaluated primarily in terms of False Rejection and False Acceptance. FR occurs when the system rejects a legitimate user and FA occurs when the system accepts an impostor as a legitimate user. Our evaluation was done using 10-fold cross-validation. The dataset was randomly sorted and we allocated in each (validation) round 90% of the dataset for training and the remaining 10% for testing; the validation results were then averaged over the different rounds.

During the enrolment mode in each round (of the cross-validation), a reference profile was generated for each user. The reference profile of the user U is based on a training set consisting of samples from the user (i.e. positive samples) and samples from other users (i.e. negative samples) considered as impostors. From the samples, we extracted a vector of features and then applied the merging and selection processes.

The verification mode is a 1-to-1 matching process and consists of comparing a sample against the enrolled user profile. FR was computed by comparing the test samples of each user U against his own profile. The FRR was obtained as the ratio between the number of false rejections and the total number of trials. FA was computed by comparing for each user U all the negative test samples against his profile. The FAR was obtained as the ratio between the number of false acceptances and the total number of trials. The overall FRR and FAR were obtained by averaging the individual measures over the entire user population. Finally, we determined the EER, which corresponds to the operating point where FRR and FAR have the same value.

We calculated the confidence of our framework using the method proposed by Bengio and Mariethoz [12] and used the HTER to calculate the CI for our system.

7.3 Evaluation Results

In this section, we start by evaluating our approach using the Twitter dataset and present baseline experimental results focusing on the different components of our approach using the same dataset and configuration. We then conduct further evaluation of our approach using the other two remaining datasets considered in this work. The list of all the features used in this experiment is shown in Table 6.1.

7.3.1 Using the Micro Messages Corpus

Using the Twitter dataset, we conducted initially a series of experiments to evaluate our proposed approach, and then performed further experiments to compare our approach against baseline methods.

Table 7.1 shows the evaluation results for our proposed approach using the Twitter dataset involving 100 authors. We started our evaluation by testing a block size of 280 characters and then reduced this subsequently to 140 characters per block, with 50, 100 and 200 blocks per users. For each test, we calculated the EER for the optimal ve limit. The best result on the Twitter dataset was achieved with block size of 280 characters and 100 blocks per user with EER of 10.08%. With this configuration, we obtain HTER = 10.06% with standard deviation $\sigma = 0.0503$. The confidence intervals calculated around this HTER for different confidence levels are listed in Table 7.2. The confidence interval around an HTER is $HTER \pm E$, where E is the margin of error.

Table 7.1: Authorship verification using DBN classifier on the Twitter dataset

Block Size	Blocks per user	ve	EER %
140	100	9.1	16.73
	200	8.8	16.58
280	50	19.0	12.61
	100	7.0	10.08

EER for the Gaussian-Bernoulli DBN classifier using the Twitter dataset involving 100 authors. In the pre-training phase, the epoch was set to 100 and the learning rate was set to 0.001. In the fine-tuning phase, the learning rate was set to 0.01.

In our baseline experiments, we analyzed the effect of modelling DBN using real-valued data versus binary data. The baseline experiments were conducted using the Twitter dataset involving 100 authors, with a block size of 280 characters and 100 blocks per user. The validation error ve (the DBN training validation error) was set to 7.0 (which corresponds to the best results in Table 7.1). Experiments when using Bernoulli-Bernoulli RBM and Gaussian-Bernoulli RBM for the visible layer yielded EER of 11.48% and 10.08%, respectively. These results show that Gaussian-Bernoulli RBM outperforms Bernoulli-Bernoulli RBM.

The above baseline experimental results indicate that our proposed approach using DBN allows a reduction of the error rate by 5.34% on average, when compared with experiments using LR classifier presented in Chapter 6 (see Table 6.4). We present in the remaining of this section, results obtained by evaluating our approach using other datasets, namely, the Enron e-mail and the forgery corpuses.

7.3.2 Using the E-mail Corpus

Our experiment using the Enron corpus was performed with a block size of 500 characters and 50 blocks or instances per user. Figure 7.3 illustrates the relationship between FRR and FAR for different values of ve varying from 0 to 50. The optimal performance achieved by our system was obtained when setting the ve limit to 15, with a FRR of 8.24% and a FAR of 8.20%. The EER was calculated as 8.21%. The HTER was found to be 8.22% with standard deviation $\sigma = 0.0648$.

Table 7.2 lists the margin of error at different confidence levels δ for the above performance value. The results show an improvement of 5.8% when compared with experiments using shallow classifiers from Chapter 6 (see Table 6.3).

7.3.3 Using the Forgery Corpus

We evaluated the robustness of our approach against the threat of forgery by simulating an adversary having access to writing samples of a user. We performed a set of experiments on the forgery corpus with 10 attackers.

For each of the 10 legal users, we calculated the FRR as explained earlier, by

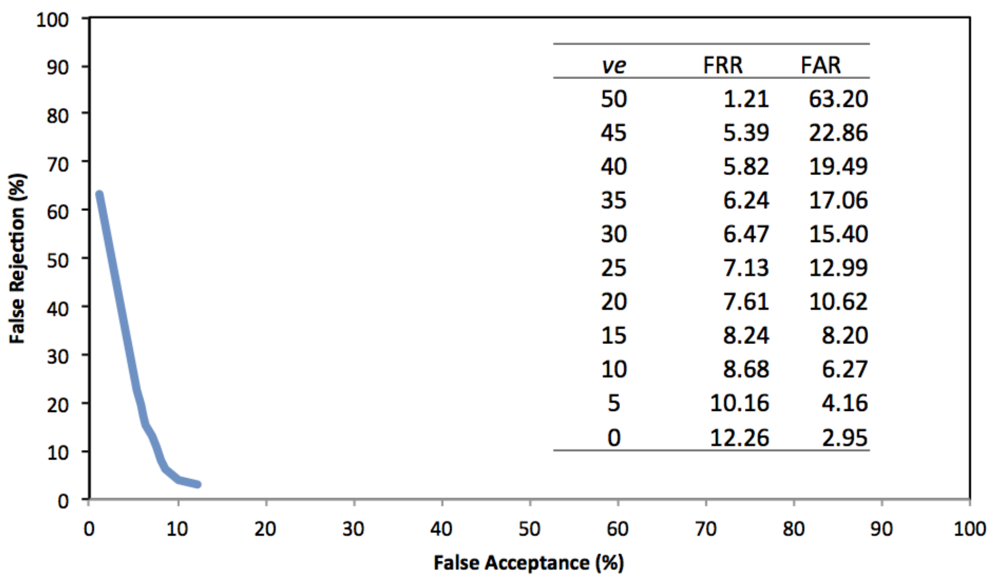


Figure 7.3: Receiver Operating Characteristic curve for the Gaussian-Bernoulli DBN classifier on the Enron corpus and sample performance values for different ve

Table 7.2: Margin of error (E) for the confidence interval for HTER Performance; δ is the confidence level

δ	E		
	Enron	Twitter	Forgery
90%	0.4123	0.2262	0.2083
95%	0.4352	0.2388	0.2198
99%	0.4535	0.2489	0.2291

HTER confidence interval for block size of 500 characters on the Enron dataset and 280 characters on the Twitter and Forgery datasets.

evaluating their own test samples against their profiles. Then we calculated the FAR by testing the 10 forgery samples of each legal user against their profile. We calculated the EER values considering the ve presented in Table 7.1.

Table 7.3 shows the obtained EER performance for 2 different block sizes, 280 and 140 characters, which are 5.48% and 12.30%, respectively. The half total error rate for block size of 280 characters was calculated as HTER = 6.68% with standard deviation $\sigma = 0.0485$. The corresponding confidence intervals for different confidence levels are shown in Table 7.2. These results indicate that the forgery attack has limited impact on the performance of the proposed method. On the other hand it can be noted that the error rates achieved for the forgery dataset are lower than the rates obtained in the previous experiments. Intuitively, such difference in performance can be explained by the fact that the forgery dataset is much smaller than the dataset used previously. The literature shows that stylometric experiments on small number of users tend to achieve better results.

Table 7.3: Authorship verification using the Forgery dataset

Block Size	Blocks per user	ve	EER%
140	100	9.1	12.30
280	100	7.0	5.48

Experiment results on the Forgery dataset involving 10 forgery attempts against 10 authors profiles.

7.4 Summary

This chapter assesses the ability of our proposed approach to address the high verification accuracy, and the ability to withstand forgery. In order to achieve the high verification accuracy, we used Deep Belief Network classifier and evaluate the DBN using Enron and Twitter dataset. Following the improvement in the accuracy of the authorship verification with DBN, we addressed the robustness of our approach against forgery attacks. The results on forgery dataset shows that an attacker as limited impact on the performance of the proposed method.

The next chapter discusses and analyzes the overall results obtained in this work.

Chapter 8

Discussions

Continuous authentication consists of monitoring the user behavior during a computing session, verifying user identity repeatedly, while discriminating between normal and suspicious user behavior. The experimental evaluation presented in the previous chapters assesses the ability of our continuous authentication model to address three key challenges related to continuous authentication: the need for short authentication delay, high verification accuracy, and the ability to withstand forgery. In this chapter, we analyse and discuss from a global perspective the results obtained in these experiments.

This chapter is structured as follows. First, we discuss in Section 8.1 experimental results related to the general characteristics of the proposed approach, such as the feature family and baseline classification. Section 8.2 analyzes the results obtained in assessing the authentication delay. Section 8.3 discusses the results on verification accuracy. Section 8.4 analyses the ability to withstand forgery. Finally, we summarize the chapter in Section 8.5.

8.1 Approach

In this section, we discuss key characteristics of our general approach, specifically we focus the discussion on the feature space, feature selection, and the effect of the SVM kernel.

All the baseline experiments conducted were based on the Enron dataset with 50 blocks per user and a block size of 500 characters. The Enron dataset has previously been used not only in authorship verification [26], but also in authorship identification [2, 42, 53–55] and authorship characterization [27, 28, 54]. These previous experiments (from the literature) used a number of users ranging from 3 to 114, and achieved in the best cases EER varying from 17% to 30%.

8.1.1 Feature Space

Beyond the feature space proposed in the literature, we introduced in this work two new feature models, n-gram model and feature merging, which based on our experiments, achieve improved performances.

For the new n-gram model, experiments reported in Section 4.2 indicate that our proposed n-gram model (which achieves EER = 14.35%) outperforms the traditional baseline model (which achieves EER = 21.26%). Experiments undertaken on the Enron dataset by varying the characters block size, and number of blocks per user, indicate that the configuration of 50 blocks per user and a block size of 500 characters achieved better results than other configuration.

Another outcome of these experiments is that 5-grams achieve better results than 3, 4, and 6-grams for large number of blocks per user and large block size. However, we noted that a block can be classified correctly by one configuration (5-grams) and misclassified by another (6-grams), suggesting that a combination of different configurations submitted to a machine learning classifier (e.g. SVM, Logistic Regression) could improve the general results. Therefore, we added to our global feature space for the new n-gram model both 5 and 6-grams.

We also extended our global feature space by defining a new feature merging process. We performed a set of experiments comparing the impact of the merging process with and without feature merging. The proposed approach achieved an improvement of 4.7%, when compared to the baseline system (see Section 6.4), which demonstrate the viability of the proposed feature model.

8.1.2 Feature Selection

Our feature selection approach reduces the feature space dimensionality by 75% eliminating irrelevant and highly correlated attributes. However, experiments reported in Section 5.3.1 showed an improvement of only 0.25% in accuracy when using the feature selection approach. This indicates that our feature selection technique has negligible impact on the accuracy of the classifier. Similar results in the literature have been reported that SVM did not benefit from feature selection [17,86]. However, feature selection is still beneficial in terms of reduction in processing time due to the reduction in the number of features.

8.1.3 Effect of SVM Kernel

We investigated the impact of different SVM kernels on accuracy as part of the baseline experiments conducted on the Enron dataset as reported in Section 5.3.1. The outcome of such study is that SVM with linear kernel achieves better EER performances than polynomial or Gaussian. The obtained results further validate the fact that the prediction accuracy of the SVM classifier can be improved by extending it with LR in an hybrid classifier.

8.2 Short Authentication Delay

We simulated short authentication delays by investigating short blocks of text. We started by investigating block sizes of 250 and 500 characters using the Enron e-mail dataset. These represent significantly shorter messages compared to the messages used so far in the literature for identity verification. To our knowledge, one of the few works that have investigated comparable message sizes includes the work by Sanderson and Guenter [90], who split a long text in chunks of 500 characters. They achieved similar results using block size of 500 characters, although with a relatively smaller dataset (i.e. 50 users). Furthermore, it is important to mention that their dataset consisted of newspapers' articles, which are known to be well structured compared to e-mail messages.

We were able to investigate even shorter messages by creating a micro-messages corpus based on Twitter feeds. As presented in Section 5.3.3, we examined messages with 140 and 280 characters per block of text. The tests have shown that 280 characters per block achieved better result than 140 characters per block. We also investigated different size of blocks per user consisting of 50, 100 and 200 blocks per user. Our results corroborate the past findings that increasing the block size and the number of blocks per user also increases the accuracy of the system.

Although the analyzed blocks of text were short, we will still need in the future to investigate even shorter messages (e.g. 10 to 50 characters) to be able to cover (beyond emails and Twitter) a broader range of online messages such as text messages (e.g. SMS, WhatsApp). However, attempting to reduce at the same time the block size and verification error rates is a difficult task in the sense that these attributes are closely related to each other. A smaller verification block may lead to increase verification error rates and vice-versa.

8.3 High Verification Accuracy

In order to investigate improvements to the verification accuracy, we examined the benefits of using a variety of shallow and deep classifiers.

8.3.1 Shallow Classifiers

We started our experiments involving shallow classifiers using the Enron and Twitter datasets, and used different configurations for block size and number of blocks per user. Table 8.1 depicts the improvement in accuracy for SVM-LR and LR over the SVM baseline classifier, based on results from Table 5.4 and 5.5. SVM-LR and LR achieve on average 9.89% and 18.62% improvement in accuracy (i.e. EER) over SVM, respectively.

Table 8.1: Accuracy improvement for SVM-LR and LR over the SVM baseline classifier

Dataset	Block size	Blocks per user	Improvement (%)	
			SVM-LR	LR
Enron	50	500	17.18	23.82
Twitter	140	100	8.68	18.90
		200	9.37	17.41
	280	50	3.47	12.51
		100	10.76	20.44

8.3.2 DBN Classifier

Compared to the existing literature, it can be claimed that the use of a machine learning method based on deep structure, specifically Deep Belief Network, helps enhance the accuracy of authorship verification using stylometry. In the early stages of our research, we investigated the standard DBN, which has binary neurons only. Our first approach was to normalize each input variable to binary values and run the

DBN classifier. However, the obtained results did not improve when compared with our previous work using a shallow structure. In order to strengthen the accuracy, we replaced the first Bernoulli-Bernoulli RBM layer for a Gaussian-Bernoulli RBM layer, which uses Gaussian units in the visible layer to model real-valued data.

Using the Enron dataset, comparing the results from Tables 6.3 and 7.1, DBN achieves on average 12.2% and 5.8% improvement in accuracy over SVM-LR and LR, respectively. Using the Twitter dataset, comparing the results from Tables 6.4 and 7.1, DBN achieves on average 7.68% and 5.34% improvement in accuracy (i.e. EER) over SVM-LR and LR, respectively. Although the results for DBN are very promising, there is still a need to improve them in order to be comparable with other biometric systems currently used for continuous authentication (e.g. keystroke and mouse dynamics). An option could be to increase the size of the block of characters, but this goes against the need for "short authentication delay" in continuous authentication. Our future work will work on this challenge.

8.4 Ability to Withstand Forgery

Stylometry analysis can be the target of forgery attacks. An adversary having access to writing samples of a user may be able to effectively reproduce many of the existing stylometric features. Section 7.3.3 showed the impact of forgery attacks on the proposed approach. The performance results obtained in our study are very encouraging. However, it is important to highlight the fact that our forgery study involved only 10 attack instances on 10 different user profiles. More data should be collected and analyzed to confirm these results, as we intend to do in the future.

8.5 Summary

This chapter summarized the experimental results obtained in evaluating our proposed approach for continuous authentication based on stylometry. We discussed the need for short authentication delay, high verification accuracy, and the ability to withstand forgery attacks. Our experiments show that our approach outperforms previous approaches in the literature. However, more work should be done in order to improve the accuracy of our stylometric model to level comparable to behavioral biometric technologies such as keystroke and mouse dynamics which are commonly used for continuous authentication.

The final chapter of the dissertation provides the conclusions of the study, a work summary, and also suggests some possible extensions for future work.

Chapter 9

Conclusion

Continuous authentication is a reinforcement of traditional static authentication (at login time) which protects against session hijacking. Continuous authentication consists of re-authenticating the user repeatedly and transparently throughout the lifetime of a computing session. The central claim of this dissertation is that continuous authentication can be accomplished through stylometric authorship verification. Our research investigated the three main challenges faced by any continuous authentication system, namely short authentication delay, authentication accuracy, and resilience to forgery. A new framework for continuous authentication using stylometry analysis has been implemented and empirically tested to support this claim.

9.1 Work Summary

Stylometry has been broadly used for authorship verification and characterization, but only a small number of works have targeted authorship verification. Our work distinguishes from previous work in this area by focusing on the challenges involved in stylometric authorship verification in the context of continuous or repeated user authentication. Most of the stylometry analysis approaches proposed in the literature uses relatively large document size, which is unacceptable for continuous authentication. Continuous authentication requires analyzing short and unstructured block of texts while keeping at the same time low verification error rates. Continuous authentication was simulated by decomposing an online text into blocks of short text. Stylometric analysis using short messages is challenging because of the limited amount of information available for decision making. Short authentication delay was achieved by investigating block sizes of 500, 280, and 140 characters.

Keeping in mind that a representative set of features could affect machine learning classification, we investigated new stylometric features. Our feature set consisted in the first place of existing lexical, syntactic, and application specific features. In addition, the framework introduces new stylometric features based on n-gram analysis and features merging. In order to select the best set of features to represent individual user profile, we computed and analyzed the information gain. Also, we applied mutual information feature selection in order to discard features that are highly correlated.

An acceptable authentication accuracy was achieved by using deep learning classifiers. Results showed that DBN classifier outperforms SVM, SVM-LR and logistic regression classifiers. Comprehensive experiments based on Enron and Twitter datasets involving 76 and 100 different authors demonstrated that the proposed approach achieves promising results when compared to existing work in the literature. The results obtained from our experimental evaluation using the Enron and Twitter datasets, consist of EER of 8.21% and 10.08% for block sizes of 500 and 280 characters, respectively. To the best of our knowledge, this is the first time that deep machine learning technique is used for the classification of stylometric profiles.

Finally, we investigated another important aspect of continuous authentication based on stylometry, which consists of resilience to forgeries. As part of this work, a novel forgery dataset was created. The evaluation of the approach using a relatively small forgery dataset yields EER varying from 5.48% to 12.3%, for different block size.

9.2 Future Work

Although the obtained results are very promising, more work should be done in the future to improve accuracy by decreasing the EER, and also by investigating shorter authentication delays (e.g. 50 characters and below). Furthermore, there is a need to confirm the results obtained in our forgery study and investigate further the resilience of our approach to forgery by expanding the dataset used. In addition, it will be interesting in the future to evaluate the proposed method with other published corpora such as the PAN[1]-2013 [60] and PAN-2014 [95] datasets.

Furthermore, with the increasing popularity of messenger services such as WhatsApp, Facebook messenger as well as Twitter, the threat of spoofing has become a source of increasing concerns for users. Future work should investigate how to extend and apply the proposed model as a spoofing counter measure. In addition, a planned future work is to investigate stylometry on mobile devices for CA purposes by analysing and extracting new stylometric features, and investigating other types of deep nets (e.g., stochastic autoencoders).

Although the experiments conducted in this work used only English-based datasets, our model can be applied for different languages with a slight adjustment to the feature selection, especially for the language dependent features (e.g. functional words). However, addressing the language multiplicity is an important issue to tackle in our future works.

[1]PAN is acronym for Uncovering Plagiarism, Authorship, and Social Software Misuse

Bibliography

[1] Ahmed Abbasi and Hsinchun Chen. Applying authorship analysis to extremist-group web forum messages. *IEEE Intelligent Systems*, 20:67–75, September 2005.

[2] Ahmed Abbasi and Hsinchun Chen. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Trans. Inf. Syst.*, 26:1–29, April 2008.

[3] A.A.E. Ahmed and I. Traore. Anomaly intrusion detection based on biometrics. In *Information Assurance Workshop, 2005. IAW '05. Proceedings from the Sixth Annual IEEE SMC*, pages 452–453, 2005.

[4] A.A.E. Ahmed and I. Traore. A new biometric technology based on mouse dynamics. *Dependable and Secure Computing, IEEE Transactions on*, 4(3):165–179, 2007.

[5] Ahmed Awad El Sayed Ahmed. *Security monitoring through human computer interaction devices.* PhD thesis, University of Victoria, 2008.

[6] Salha M. Alzahrani, Naomie Salim, and Ajith Abraham. Understanding plagiarism linguistic patterns, textual features, and detection methods. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(2):133–149, March 2012.

[7] Shlomo Argamon, Casey Whitelaw, Paul Chase, Sobhan Raj Hota, Navendu Garg, and Shlomo Levitan. Stylistic text classification using functional lexical features: Research articles. *J. Am. Soc. Inf. Sci. Technol.*, 58:802–822, April 2007.

[8] Harald Baayen, Hans van Halteren, and Fiona Tweedie. Outside the cave of shadows: using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing*, 11(3):121–132, 1996.

[9] Eric Backer and Peter van Kranenburg. On musical stylometry pattern recognition approach. *Pattern Recognition Letters*, 26(3):299–309, 2005.

[10] Lucas Ballard. Biometric authentication revisited: Understanding the impact of wolves in sheep's clothing. In *In Proceedings of the 15 th Annual Usenix Security Symposium*, pages 29–41, 2006.

[11] Ricardo Barandela, Rosa M. Valdovinos, J. Salvador Sanchez, and Francesc J. Ferri. The imbalanced training sample problem: Under or over sampling? In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 806–814. Springer, 2004.

[12] Samy Bengio and Johnny Mariethoz. A statistical significance test for person authentication. In *ODYSSEY04-The Speaker and Language Recognition Workshop*, 2004.

[13] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.

[14] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[15] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[16] Y-lan Boureau, Yann L Cun, et al. Sparse feature learning for deep belief networks. In *Advances in neural information processing systems*, pages 1185–1192, 2008.

[17] Janez Brank, Marko Grobelnik, Natasa Milic-Frayling, and Dunja Mladenic. Interaction of feature selection methods and linear classification models. In *Workshop on Text Learning held at ICML*, 2002.

[18] Marcelo Luiz Brocardo, Issa Traore, Sherif Saad, and Isaac Woungang. Authorship verification for short messages using stylometry. In *In Proceedings of the International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–6. Piraeus-Athens, Greece, May 2013.

[19] John Burrows. Delta: a measure of stylistic difference and a guide to likely authorship. *Literary and Linguistic Computing*, 17(3):267–287, 2002.

[20] John F Burrows. Word patterns and story shapes: The statistical analysis of narrative style. *Literary and Linguistic Computing*, 2(1):61–70, 1987.

[21] Jie Cai, Yuezhong Tang, and Rile Hu. Spam filter for short messages using winnow. In *Advanced Language Processing and Web Information Technology, 2008. ALPIT'08. International Conference on*, pages 454–459. IEEE, 2008.

[22] Fazli Can and Jon M. Patton. *Change of Writing Style With Time*, volume 38. Kluwer Academic Publishers, 2004.

[23] Omar Canales, Vinnie Monaco, Thomas Murphy, Edyta Zych, John Stewart, Charles Tappert Alex Castro, Ola Sotoye, Linda Torres, and Greg Truley. A stylometry system for authenticating students taking online tests. CSIS, Pace University, May 6 2011.

[24] Tanmoy Chakraborty. Authorship identification using stylometry analysis in bengali literature. *CoRR*, 2012.

[25] Carole E. Chaski. Who's at the keyboard: Authorship attribution in digital evidence investigations. *International Journal of Digital Evidence*, 4(1):1–13, Spring 2005.

[26] Xiaoling Chen, Peng Hao, R. Chandramouli, and K. P. Subbalakshmi. Authorship similarity detection from email messages. In *Proceedings of the 7th international conference on Machine learning and data mining in pattern recognition*, MLDM'11, pages 375–386, Berlin, Heidelberg, 2011. Springer-Verlag.

[27] Na Cheng, R. Chandramouli, and K.P. Subbalakshmi. Author gender identification from text. *Digital Investigation*, 8(1):78–88, 2011.

[28] Na Cheng, Xiaoling Chen, R. Chandramouli, and K.P. Subbalakshmi. Gender identification from e-mails. In *Computational Intelligence and Data Mining, 2009. CIDM '09. IEEE Symposium on*, pages 154–158, 30 2009-april 2 2009.

[29] Yuan chin Ivan Chang. Boosting svm classifiers with logistic regression. Technical report, Institute of Statistical Science - Academia Sinica, Taipei, Taiwan, 03 2003.

[30] Kyung Hyun Cho, Tapani Raiko, and Alexander Ilin. Gaussian-bernoulli deep boltzmann machine. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–7. IEEE, 2013.

[31] Jonathan H. Clark and Charles J. Hannon. A classifier system for author recognition using synonym-based features. In *Proceedings of the 6th Mexican international conference on Advances in artificial intelligence*, MICAI'07, pages 839–849, Berlin, Heidelberg, 2007. Springer-Verlag.

[32] Malcolm Corney, Olivier de Vel, Alison Anderson, and George Mohay. Gender-preferential text mining of e-mail discourse. In *Proceedings of the 18th Annual Computer Security Applications Conference*, pages 282–289, 2002.

[33] Olivier de Vel, Alison Anderson, Malcolm Corney, and George Mohay. Mining e-mail content for author identification forensics. *Sigmod Record*, 30(4):55–64, 2001.

[34] Li Deng, Ossama Abdel-Hamid, and Dong Yu. A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6669–6673. IEEE, 2013.

[35] Wei-Wei Deng and Hong Peng. Research on a naive bayesian based short message filtering system. In *Machine Learning and Cybernetics, 2006 International Conference on*, pages 1233–1237, 2006.

[36] Ronald Doyle, John Hind, and Marcia Peters. Technique for continuous user authentication - patent us 20020095586 a1. In *International Business Machines Corporation*, 2001.

[37] Benoit Duc, Stefan Fischer, and Josef Bigun. Face authentication with gabor information on deformable graphs. *Image Processing, IEEE Transactions on*, 8(4):504–516, 1999.

[38] Usama M. Fayyad and Keki B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Thirteenth International Joint Conference on Articial Intelligence*, volume 2, pages 1022–1027. Morgan Kaufmann Publishers, 1993.

[39] Davrondzhon Gafurov and Einar Snekkenes. Gait recognition using wearable motion recording sensors. *EURASIP Journal on Advances in Signal Processing*, 2009:7, 2009.

[40] Michael Gamon. Linguistic correlates of style: authorship classification with deep linguistic analysis features. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[41] Sonia Garcia-Salicetti, Charles Beumier, Gérard Chollet, Bernadette Dorizzi, Jean Leroux les Jardins, Jan Lunter, Yang Ni, and Dijana Petrovska-Delacrétaz. Biomet: A multimodal person authentication database including face, voice, fingerprint, hand and signature modalities. In *Audio-and Video-Based Biometric Person Authentication*, pages 845–853. Springer, 2003.

[42] Rachid Hadjidj, Mourad Debbabi, Hakim Lounis, Farkhund Iqbal, Adam Szporer, and Djamel Benredjem. Towards an integrated e-mail forensic analysis framework. *Digital Investigation*, 5(3-4):124–137, 2009.

[43] Hans Van Halteren. Author verification by linguistic profiling: An exploration of the parameter space. *ACM Trans. Speech Lang. Process.*, 4:1–17, February 2007.

[44] John. L. Hilton. *On verifying wordprint studies: Book of Mormon authorship*. Reprint (Foundation for Ancient Research and Mormon Studies). F.A.R.M.S., 1991.

[45] Geoffrey Hinton. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1), 2010.

[46] Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[47] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[48] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[49] N. Homem and J.P. Carvalho. Authorship identification and author fuzzy fingerprints. In *Fuzzy Information Processing Society (NAFIPS), 2011 Annual Meeting of the North American*, pages 1–6, march 2011.

[50] David I. Homes. The evolution of stylometry in humanities scholarship. *Literary and Linguistic Computing*, 13(3):111–117, 1998.

[51] Donato Impedovo and Giuseppe Pirlo. Automatic signature verification: the state of the art. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(5):609–635, 2008.

[52] Giacomo Inches and Fabio Crestani. Online conversation mining for author characterization and topic identification. In *Proceedings of the 4th workshop on Workshop for Ph.D. students in information e knowledge management*, PIKM '11, pages 19–26, New York, NY, USA, 2011. ACM.

[53] Farkhund Iqbal, Hamad Binsalleeh, Benjamin C.M. Fung, and Mourad Debbabi. Mining writeprints from anonymous e-mails for forensic investigation. *Digital Investigation*, 7(1-2):56–64, 2010.

[54] Farkhund Iqbal, Hamad Binsalleeh, Benjamin C.M. Fung, and Mourad Debbabi. A unified data mining solution for authorship analysis in anonymous textual communications. *Information Sciences*, 231:98–112, 2013.

[55] Farkhund Iqbal, Rachid Hadjidj, Benjamin C.M. Fung, and Mourad Debbabi. A novel approach of mining write-prints for authorship attribution in e-mail forensics. *Digital Investigation*, 5:S42–S51, 2008.

[56] Farkhund Iqbal, Liaquat A. Khan, Benjamin C. M. Fung, and Mourad Debbabi. E-mail authorship verification for forensic investigation. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 1591–1598, New York, NY, USA, 2010. ACM.

[57] Anil K Jain, Ruud Bolle, and Sharath Pankanti. *Biometrics: personal identification in networked society*. Springer, 1999.

[58] Patrick Juola. Authorship attribution for electronic documents. In *Advances in Digital Forensics II*, volume 222 of *IFIP Advances in Information and Communication*, pages 119–130. Springer New York, 2006.

[59] Patrick Juola and R. Harald Baayen. A controlled-corpus experiment in authorship identification by cross-entropy. *Literary and Linguistic Computing*, 20(Suppl):59–67, 2005.

[60] Patrick Juola and Efstathios Stamatatos. Overview of the author identification task at pan 2013. In *Conference and Labs of the Evaluation Forum - CLEF 2013*, Valencia - Spain, 2013.

[61] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, second edition, February 2008.

[62] Jussi Karlgren and Gunnar Eriksson. Authors, genre, and linguistic convention. In *SIGIR '07 Amsterdam. Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection*, 2007.

[63] Bradley Kjell, W.Addison Woods, and Ophir Frieder. Discrimination of authorship using visualization. *Information Processing and Management*, 30(1):141–150, 1994.

[64] Bryan Klimt and Yiming Yang. The enron corpus: A new dataset for email classification research. In *Machine learning: ECML 2004*, pages 217–226. Springer, 2004.

[65] Moshe Koppel and Jonathan Schler. Exploiting stylistic idiosyncrasies for authorship attribution. In *IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis*, pages 69–72, Acapulco, Mexico, 2003.

[66] Moshe Koppel and Jonathan Schler. Authorship verification as a one-class classification problem. In *Proceedings of the 21st international conference on Machine learning*, ICML '04, pages 62–69, Banff, Alberta, Canada, 2004. ACM.

[67] Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Computational methods in authorship attribution. *J. Am. Soc. Inf. Sci. Technol.*, 60:9–26, January 2009.

[68] Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Authorship attribution in the wild. *Lang. Resour. Eval.*, 45:83–94, March 2010.

[69] Moshe Koppel and Yaron Winter. Determining if two documents are written by the same author. *Journal of the Association for Information Science and Technology*, 65(1):178–187, 2014.

[70] Sotiris Kotsiantis and Dimitris Kanellopoulos. Discretization techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering*, 32(1):47–58, 2006.

[71] Oswin Krause, Asja Fischer, Tobias Glasmachers, and Christian Igel. Approximation properties of dbns with binary hidden units and real-valued visible units. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 419–426, 2013.

[72] Ivan Krsul and Eugene H. Spafford. Authorship analysis: identifying the author of a program. *Computers and Security*, 16(3):233–257, 1997.

[73] Tayfun Kucukyilmaz, B. Barla Cambazoglu, Cevdet Aykanat, and Fazli Can. Chat mining: Predicting user and message attributes in computer-mediated communication. *Information Processing Management*, 44(4):1448–1466, 2008.

[74] Niels Landwehr, Mark Hall, and Eibe Frank. Logistic model trees. *Mach. Learn.*, 59(1-2):161–205, May 2005.

[75] L. Latha and S. Thangasamy. A robust person authentication system based on score level fusion of left and right irises and retinal features. *Procedia Computer Science*, 2(0):111 – 120, 2010. Proceedings of the International Conference and Exhibition on Biometrics Technology.

[76] Jiexun Li, Rong Zheng, and Hsinchun Chen. From fingerprint to writeprint. *Commun. ACM*, 49:76–82, April 2006.

[77] Philip M. McCarthy, Gwyneth A. Lewis, David F. Dufty, and Danielle S. Mcnamara. Analyzing writing styles with coh-metrix. In *Proceedings of the Florida Artificial Intelligence Research Society International Conference (FLAIRS)*, 2006.

[78] Thomas Corwin Mendenhall. The characteristic curves of composition. *Science*, (214S):237–246, 1887.

[79] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41, 1995.

[80] Frederick Mosteller and David L Wallace. *Inference and Disputed Authorship: The Federalist*. Addison-Wesley, 1964.

[81] Arvind Narayanan, Hristo Paskov, Neil Zhenqiang Gong, John Bethencourt, Emil Stefanov, Eui Chul Richard Shin, and Dawn Song. On the feasibility of internet-scale author identification. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, SP '12, pages 300–314, Washington, DC, USA, 2012. IEEE Computer Society.

[82] Angela Orebaugh and Jeremy Allnutt. Classification of instant messaging communications for forensics analysis. *The International Journal of Forensic Computer Science*, 1:22–28, 2009.

[83] D. Pavelec, L.S. Oliveira, E. Justino, F.D.N. Neto, and L.V. Batista. Author identification using compression models. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 936–940, july 2009.

[84] Fuchun Peng, Dale Schuurmans, Shaojun Wang, and Vlado Keselj. Language independent authorship attribution using character level language models. In *Proceedings of the 10th Conference on European. Chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 267–274, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[85] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

[86] Monica Rogati and Yiming Yang. High-performing feature selection for text classification. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 659–661. ACM, 2002.

[87] Kailash Gajulapalli Ruchita Sarawgi and Yejin Choi. Gender attribution: tracing stylometric evidence beyond topic and genre. In *Proceedings of the 15th Conference on Computational Natural Language Learning*, CoNLL '11, pages 78–86, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[88] Tara N Sainath, Brian Kingsbury, and Bhuvana Ramabhadran. Auto-encoder bottleneck features using deep belief networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4153–4156. IEEE, 2012.

[89] Conrad Sanderson and Simon Guenter. On authorship attribution via markov chains and sequence kernels. In *Proceedings of the 18th International Conference on Pattern Recognition - Volume 03*, ICPR '06, pages 437–440, Washington, DC, USA, 2006. IEEE Computer Society.

[90] Conrad Sanderson and Simon Guenter. Short text authorship attribution via sequence kernels, markov chains and author unmasking: an investigation. In

Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP '06, pages 482–491, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[91] Ruhi Sarikaya, Geoffrey E Hinton, and Bhuvana Ramabhadran. Deep belief nets for natural language call-routing. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5680–5683. IEEE, 2011.

[92] Paul Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In MIT Press, editor, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 194–281. Department of Computer Science, University of Colorado, Boulder, 1986.

[93] Efstathios Stamatatos. Ensemble-based author identification using character n-grams. In *3rd International Workshop on Text-based Information Retrieval*, pages 41–46, 2006.

[94] Efstathios Stamatatos. A survey of modern authorship attribution methods. *J. Am. Soc. Inf. Sci. Technol.*, 60:538–556, March 2009.

[95] Efstathios Stamatatos, Walter Daelemans, Ben Verhoeven, Martin Potthast, Benno Stein, Patrick Juola, Miguel A Sanchez-Perez, and Alberto Barrón-Cedeño. Overview of the author identification task at pan 2014. In *Conference and Labs of the Evaluation Forum - CLEF 2014*, Sheffield - UK, 2014.

[96] Efstathios Stamatatos and Moshe Koppel. Plagiarism and authorship analysis: introduction to the special issue. *Language Resources and Evaluation*, 45:1–4, 2011. 10.1007/s10579-011-9136-1.

[97] Benno Stein, Nedim Lipka, and Peter Prettenhofer. Intrinsic plagiarism analysis. *Language Resources and Evaluation*, 45:63–82, 2011.

[98] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

[99] Issa Traore and Ahmed Awad E. Ahmed, editors. *Continuous Authentication Using Biometrics: Continuous Authentication Using Biometrics: Data, Models, and Metrics*. IGI Global, 2012.

[100] Vladimir Vapnik. *Estimation of Dependences Based on Empirical Data: Springer Series in Statistics (Springer Series in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982.

[101] Olivier De Vel. Mining e-mail authorship. In *Proc. Workshop on Text Mining, ACM International Conference on Knowledge Discovery and Data Mining (KDD'2000)*, 2000.

[102] Grace Wahba, Grace Wahba, Xiwu Lin, Xiwu Lin, Fangyu Gao, Fangyu Gao, Dong Xiang, Dong Xiang, Ronald Klein, and Barbara Klein. The bias-variance tradeoff and the randomized gacv. In *Advances in Neural Information Processing Systems*, pages 620–626. MIT Press, 1999.

[103] Carrington B Williams. A note on the statistical analysis of sentence-length as a criterion of literary style. *Biometrika*, pages 356–361, 1940.

[104] Carrington B Williams. Mendenhall's studies of word-length distribution in the works of shakespeare and bacon. *Biometrika*, 62(1):207–212, 1975.

[105] Ian H. Witten, Eibe Frank, Len Trigg, Mark Hall, Geoffrey Holmes, and Sally Jo Cunningham. Weka: Practical machine learning tools and techniques with java implementations. In *ANNES'99 International Workshop on emerging Engineering and Connectionnist-based Information Systems*, pages 192–196, 1999.

[106] George Udny Yule. On sentence-length as a statistical characteristic of style in prose: with application to two cases of disputed authorship. *Biometrika*, 30(3-4):363–390, 1939.

[107] Ying Zhao and Justin Zobel. Searching with style: authorship attribution in classic literature. In *Proceedings of the thirtieth Australasian conference on Computer science - Volume 62*, ACSC '07, pages 59–68, Darlinghurst, Australia, Australia, 2007. Australian Computer Society, Inc.

[108] Rong Zheng, Jiexun Li, Hsinchun Chen, and Zan Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. *J. Am. Soc. Inf. Sci. Technol.*, 57:378–393, February 2006.

[109] Guanyu Zhou, Kihyuk Sohn, and Honglak Lee. Online incremental feature learning with denoising autoencoders. In *International Conference on Artificial Intelligence and Statistics*, pages 1453–1461, 2012.

[110] George Kingsley Zipf. *Selected Studies of the Principle of Relative Frequency in Language*. Harvard University Press, 1932.