Messaging Forensic Framework for Cybercrime Investigation

Farkhund Iqbal

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montréal, Québec, Canada

January 2011

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Farkhund Iqbal**

Entitled: **Messaging Forensic Framework for Cybercrime Investigation**

and submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. C. Mulligan

_____ External Examiner
Dr. W. K.-W. Cheung

_____ Examiner to Program
Dr. A. Agarwal

_____ Examiner
Dr. O. Ormandjieva

_____ Examiner
Dr. C. Y. Suen

_____ Thesis Co-Supervisor
Dr. M. Debbabi

_____ Thesis Co-Supervisor
Dr. B. Fung

Approved by _____
Dr. H. Harutyunyan, Graduate Program Director

January 27, 2011 _____
Dr. Robin A.L. Drew, Dean
Faculty of Engineering and Computer Science

ABSTRACT

Messaging Forensic Framework for Cybercrime Investigation

Farkhund Iqbal, Ph. D.

Concordia University, 2011

Online predators, botmasters, and terrorists abuse the Internet and associated web technologies by conducting illegitimate activities such as bullying, phishing, and threatening. These activities often involve online messages between a criminal and a victim, or between criminals themselves. The forensic analysis of online messages to collect empirical evidence that can be used to prosecute cybercriminals in a court of law is one way to minimize most cybercrimes. The challenge is to develop innovative tools and techniques to precisely analyze large volumes of suspicious online messages. We develop a forensic analysis framework to help an investigator analyze the textual content of online messages with two main objectives. First, we apply our novel authorship analysis techniques for collecting patterns of authorial attributes to address the problem of anonymity in online communication. Second, we apply the proposed knowledge discovery and semantic analysis techniques for identifying criminal networks and their illegal activities. The focus of the framework is to collect creditable, intuitive, and interpretable evidence for both technical and non-technical professional experts including law enforcement personnel and

jury members. To evaluate our proposed methods, we share our collaborative work with a local law enforcement agency. The experimental result on real-life data suggests that the presented forensic analysis framework is effective for cybercrime investigation.

DEDICATION

To my parents who have always been affectionate to me,

To my wife, brothers, and sister who have been incredibly patient and supportive,

To my kids who have been giving me the strength through their sweet smiles.

Acknowledgements

I would like to make 'SHUKR' to my Beloved ALLAH, the most Merciful and Cherisher, who created me and then showered me with his countless bounties.

I would like to thank my supervisors Prof. Mourad Debbabi and Dr. Benjamin Fung for their indispensable and incredible guidance. Prof. Debbabi gave me well-structured research plan with clearly defined milestones while Dr. Fung helped me to precisely meet the milestones. Our research objectives would not have been achieved without the professional and experienced guidance and support of my supervisors.

My gratefulness extends to members of the examining committee including Dr. W. K.-W. Cheung, Dr. A. Agarwal, Dr. O. Ormandjieva, and Dr. C. Y. Suen for critically evaluating my thesis and giving me valuable feedback. My special thanks go to Hamad BinSalleeh, Amine Boukhetouta, Khalid Sultan, Irshad Ali, Neharullah, and Omar Mery for their nice company and encouragement. I extend my gratitude to members of our Computer Security Laboratory and to the faculty members especially Dr. Amr Youssef and Dr. A. Ben Hamza for their sincere advices. I am grateful to the staff members of CIISE for their help and assistance during my stay at Concordia University.

Finally, I take this opportunity to express my profound gratitude to my beloved parents, brothers, sister, my wife and my kids for their moral support and patience during my studies at Concordia University.

List of abbreviations

FTK	Forensic ToolKit
EMT	E-mail Mining Toolkit
FP	Frequent Patterns
WP	Writeprint
SVM	Support Vector Machine
DT	Decision Tree
END	Ensemble of Nested Dichotomies
NER	Named Entity Recognition
NLP	Natural Language Processing
DNA	DeoxyriboNucleic Acid
WEKA	Waikato Environment for Knowledge Analysis
RBFNetwork	Radial Basis Function Network
BayesNet	Bayesian Networks
GUI	Graphical User Interface
HTML	HyperText Markup Language
NB	Naive Bayes
EM	Expectation-Maximization
ARFF	Attribute-Relation File Format
UBM	Universal Background Model

EER	Equal Error Rate
DCF	Cost Detection Function
kNN	k-nearest neighbor
min_sup	Minimum Support
DET	Detection Error Trade-off
ROC	Receiver Operating Characteristic
IRC	Internet Relay Chat
mDCF	minimum Cost Detection Function
MSN	Microsoft Network
DMNB	Discriminative Multinomial Naive Bayes
SVM-SMO	Support Vector Machine with Sequential Minimum Optimization
SVM-RBF	Support Vector Machine with Radial Basis Function
KW	Keyword
CC	Common Concept
KC	Key Concept
ST	Special Terms
RS	Related Synset
OS	Overlapping Synset
TF	Term Frequency
IDF	Inverse Document Frequency
SRE	Speaker Recognition Evaluation

TABLE OF CONTENTS

FIGURES

TABLES

Chapter 1

Introduction

Cybercriminals abuse the anonymity in online communication for conducting illegitimate activities including phishing, spamming, identity theft, masquerade, threatening, and harassment. In phishing scams, for instance, scammers send out phishing messages and create phishing websites to trick account holders into disclosing their sensitive account information, such as account number and password. Similarly, the reputation systems of online marketplaces, built by using customers' feedback, is most often manipulated by entering the system with multiple names (aliases) [5]. Terrorist groups and criminal gangs use the Internet and World Wide Web for committing organized crimes such as armed robbery, drug trafficking, and acts of terror [24, 90]. They use online messaging systems as safe channels for their covert communication. The digital revolution has greatly simplified the ability to copy and distribute creative works, which has led to increased copyright violation worldwide [110].

In most Internet-mediated crimes, the victimization tactics used vary from simple anonymity to identity theft and masquerade. In distributing unsolicited junk mail, called spamming, for instance, a perpetrator attempts to hide his/her true identity, while in phishing s/he may impersonate an officer of high authority. In predatory and bullying chat conversation, a pedophile more likely pretends to be a teenager [52]. Similarly, in web spoofing [27] a potential victim is tricked (through a bulk message) into uploading personal information on a deceptive website. Likewise, in escrow fraud websites [28], a fake seller creates a dummy online escrow service and then disappears after collecting money from the buyers.

In this thesis, we develop a forensic analysis framework for analyzing online messages by integrating data mining algorithms, natural language processing techniques, and social networking analysis techniques. The developed framework can be employed to *automatically* perform a *multi-stage analysis* of suspicious online documents and present the findings with objectivity and intuitiveness. The challenge is to collect evidence that is creditable, intuitive, and is interpretable by both technical and non-technical professional experts, i.e, law enforcement personnel or jury members. The analysis can be applied to the header as well as the body of an online message.

Depicted in Figure 1.1, the header-content is analyzed to collect preliminary information about the general behavior of the users. The body-content or message body is analyzed to collect forensically relevant information about the potential suspects and their activities. The information extracted from the textual body of a message are used as *internal* evidence [64]. This thesis is focused on analyzing the message body. The term online

message is used throughout the thesis to represent the Internet-mediated communication documents including e-mails, chat logs, blogs, and forum posts.

The analysis of header-content can help an investigator collect preliminary information about the incident and thus can shape the process of an investigation. In the initial phase of an investigation, given a suspicious dataset, e.g., an e-mail corpus, an investigator may want to collect simple statistics such as e-mail distribution based on sender, recipient, and the time at which a message is sent. Similarly, an investigator may want to learn about the social behavior of the suspects within their communities and social groups by applying social networking techniques. Furthermore, identifying the physical distribution of e-mail users may unveil important information leads. We achieve this functionality by applying geographical localization and map retrieval techniques on the given message collection.

Sometimes the task of an investigator would be to classify a given message to one of the predefined topic categories. Most spam filtering and scanning systems are using topic- or content-based classification techniques. We use some topic categories with example documents to develop a classification model that is employed for identifying the topic of new messages. Sometimes the task of an investigator would be to simply identify the pertinent topics within a large collection of documents without having predefined topics. For this, we apply unsupervised learning techniques, called clustering, in our framework.

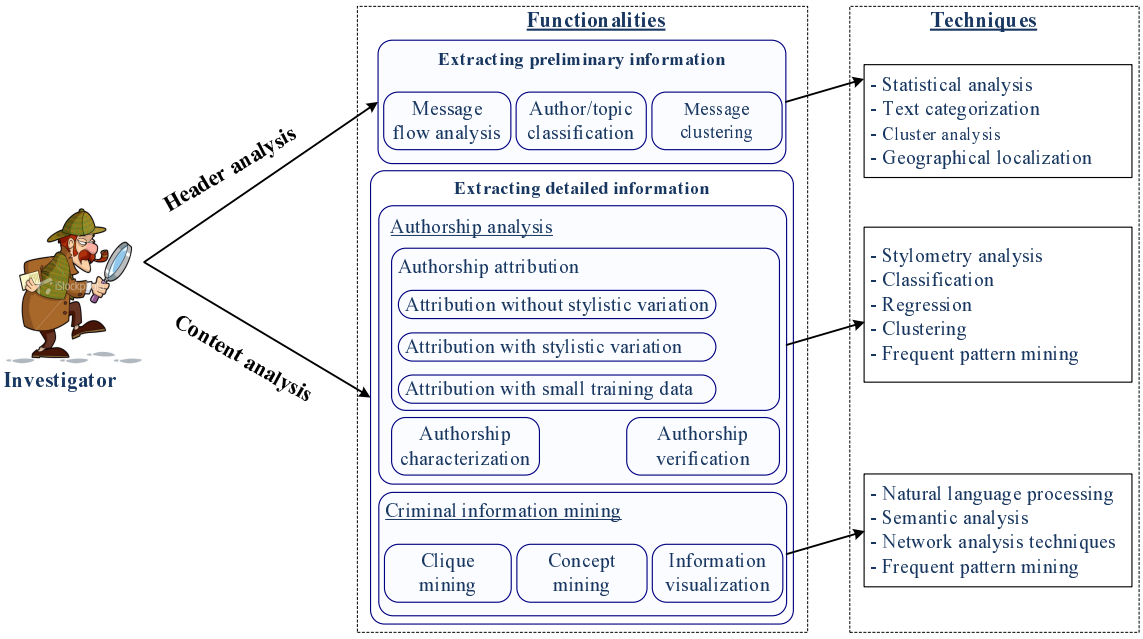


Figure 1.1: Framework overview

The presented framework analyzes the message body to: (1) collect traces of authorial attributes for addressing the anonymity issue, called authorship analysis, and (2) extract forensically relevant information, called criminal information mining, from the textual content of online messages. *Authorship analysis* is applied to extract author-specific features from the sample documents of a suspect to create his/her writeprint. The created writeprint is the combination of stylistic features that are frequently found in the sample documents of one suspect only and not in documents of other suspects. The proposed authorship approach can be applied to authorship identification, authorship verification, and authorship characterization problems (discussed in Section 1.1). *Criminal information mining* is applied to extract knowledge about potential criminal entities and their illegal activities from suspicious online documents. Finally, we use social networking techniques to present the extracted information for further investigations.

The rest of the chapter is organized as follows: Section 1.1 presents the motivation and problem statement. Section 1.2 and Section 1.3, respectively, list the objectives and contributions of the thesis. The structure of the thesis is given in Section 1.4.

1.1 Motivation and Problem Description

Installing antiviruses, filters, intrusion detection systems, and firewalls is not sufficient [104] to secure online communication. Moreover, to identify the source of a malicious message, an investigator usually needs to backtrack the IP addresses based on the information in the header of the anonymous message. However, solely based on tracking the IP address is insufficient to identify the suspect (e.g., the author of an anonymous message) if there are multiple users on the computer that sent out the message, or if the message is sent from a proxy server. In cases of hacked e-mail accounts and compromised computers, the metadata contained in the header are forged and anonymized and thus cannot be trusted. Similarly, monitoring chat rooms to detect possible predatory or bullying attacks by entering suspicious chat forums with pseudo-victim ID is not a trivial task.

In this context, forensic analysis of online messages to collect empirical evidence to prosecute an offender of a cybercrime by means of law is one way to minimize cybercrimes [104]. The large volume of online messages often contain enormous amount of forensically relevant information about potential suspects and their illegitimate activities. The existing tools, e.g., Forensic ToolKit [2], Encase [3], COPLINK solution suite [73], and Paraben e-mail examiner [1] are some general-purpose analysis software and are not

designed specifically for analyzing the textual contents of online messages. E-mail Mining Toolkit (EMT) [102], on the other hand, is a free e-mail analysis software that computes user behavioral models based on communication patterns. The toolkit is limited to analyze e-mail documents only. The aforementioned tools do not have the functionality of authorship analysis for resolving authorial disputes.

The challenge is to develop innovative tools and techniques that can be employed to collect forensic evidence by analyzing both the header and the body of an online message. The collected evidence needs to be not only precise but also intuitive, interpretable, and traceable. Header-level information, e.g., IP addresses, host names, and sender and recipient addresses, contained in an e-mail header; the user ID used in chatting: and the screen names used in web-based communication help reveal information at the user or application level. For instance, the header content extracted from a suspicious e-mail corpus helps reveal who the senders and recipients are and how often they communicate, how many types of communities there are in the dataset, and what are the inter- and intra-community patterns of communication. The body of a message can be analyzed to collect information about the potential authors and the perceived underlying semantics of the written text [12].

In this section, we briefly discuss the motivations of the current study and identify the challenges faced by an investigator in analyzing online documents. Most existing studies focus on investigating the header-content, while very few studies have been conducted on analyzing the body-content. The focus of this thesis is to formulate problems and propose solutions in the area of content-level analysis. For header analysis, we implement

existing state-of-the-art techniques including statistical analysis, geographical localization, social network analysis, and text categorization methods. A detailed description of these techniques is given in Chapter 3 of this thesis.

The textual content of a message is studied mainly from two perspectives: authorship analysis and criminal information mining. Authorship analysis is applied to address the issue of anonymity in cybercrime investigation. Knowledge discovery or criminal information mining techniques are applied to learn about the illegitimate activities of cybercriminals. Therefore, we identify the motivations and research challenges in the aforementioned two research areas and discuss them in the following sections.

1.1.1 Authorship Analysis

Most existing authorship studies employ classifiers to infer the author of anonymous documents. The classifiers, commonly used in these studies, fall into three main categories: (1) probabilistic classifiers, e.g., Bayesian classifiers [91] and its variants; (2) decision trees [87], e.g., C4.5 and J48; and (3) support vector machine [61] and its variants, e.g., Ensemble SVM. Each of these techniques has its own limitations in terms of classification accuracy, scalability, and interpretability. An extensive survey on text categorization [96] suggests that SVM outperforms most classifiers, such as decision tree methods [88, 89], the probabilistic naive Bayes classifier, and batch linear classifiers (Rocchio).

Though support vector machine outperforms most classifiers including decision trees in terms of accuracy, it is a black box approach and the results produced by this classifier are not interpretable. Therefore, it is not suitable for evidence collection. Decision

trees, on the other hand, are symbolic and not quantitative and are therefore interpretable. However, in building a decision tree, only the local information of a node is considered and therefore it fails to capture the integrated effect of different features; thus, the results are not very accurate.

The accuracy of most classifiers is subject to the size of data available for training. However, in most cybercrime investigations the example data is hardly enough to train a classifier. Similarly, most authorship studies focus on structured documents such as books, which are relatively easy to analyze as compared to unstructured data such as online messages.

We study authorship problem from the following four perspectives.

Authorship attribution. An investigator has a disputed anonymous online message together with some potential suspects. The task of the investigator is to identify the true author of the document in question by analyzing the sample documents of potential suspects. Although existing authorship studies mention temporal and contextual variation in the writing style of an author, they do not take them into consideration. In this thesis, we address the problem of authorship attribution with and without a focus on the problem of *stylistic variation*. The term stylistic variation is used to represent the temporal and occasional change in the writing style of an individual.

Authorship identification with few training samples. In most real-world investigation problems, the number of sample documents is often insufficient for training a classifier. In certain situations the available sample may be very small or there may be no sample. In some cases, an investigator can ask a suspect to produce a sample of her

writing by listening to a story or watching a movie and then reproducing the played scene in his/her own writing. Clearly, the number of samples is very limited.

Authorship characterization. Sometimes a cybercrime investigator has no clue about who the potential suspects are and therefore has no training samples. Yet, the investigator would like to infer characteristics of the author(s), such as gender, age group, and ethnic group, based on the writing styles in the anonymous messages. We assume the investigator has access to some external source of text messages such as blog postings and social network websites that disclose the authors' public profiles. The challenge is how to utilize such external sources to infer characteristics of the authors of the anonymous messages.

Authorship verification. The problem is to confirm whether or not the given disputed anonymous message is written by a given suspect. Some researchers treat verification as a similarity detection problem in which the task is to determine if the two given objects are produced by the same entity, without knowing explicitly about the entity. The need is to first clearly define the problem of authorship verification and then propose a solution.

The challenge is not only to address the aforementioned authorship problems but also to support the findings with strong evidence for forensic purposes.

1.1.2 Criminal Information Mining

In the study of authorship analysis, the task is to extract the content-independent attributes, called stylometric features, from the textual content of documents. On the other hand,

in criminal information mining, the task is to analyze the content-specific words of the documents to collect forensically relevant information. The extracted information can be used to answer questions such as: What are the pertinent suspicious entities mentioned within a document? Are these entities related to each other? What concepts and topics are discussed in the documents?

Online documents can be analyzed to reveal information about suspicious entities and their malicious activities. Identifying the semantic meaning of the written words by applying contextual analysis and disambiguation techniques will help the investigator retrieve malicious documents. Understanding the perceived (semantic) meaning of suspicious messages is not trivial due to the obfuscation and deception techniques used by perpetrators in their online communication. For instance, the perceived meaning of written words in a malicious discourse is different from their apparent meaning, as the street names used for most illegitimate activities are borrowed from daily conversation. The word 'thunder' means heroin and the word 'snow' means cocaine in e-mails used for drug trafficking. There are more than 2300 street terms (used for drugs or drug-related activities) available on `http://www.whitehousedrugpolicy.gov`.

Predictive machine learning methods and natural language-processing techniques are applied to extract this information. Named entity recognition [8] is employed to extract traces of information related to persons, locations, or objects. Social networking [24] and link analysis techniques [95] are applied to identify covert associations between entities. Similarly, topic identification or topic detection is employed to identify the topic or genre of a document [9]. Text summarization techniques [14, 15, 32, 109] are usually

employed to extract a summary of textual documents.

The limitations of most existing criminal information mining techniques are: (1) Forensic tools and techniques, e.g., COPLINK solutions suite, are used to collect network-level information, e.g., URL and host name, instead of analyzing the textual content of the documents. (2) Most analysis techniques, designed for text classification and clustering, consider only the frequency of words and not their semantics. (3) The proposed approaches focus on structured data, i.e., formal reports, rather than unstructured data such as chat logs and e-mail messages. (4) Most existing forensic tools are either developed for very high level analysis, e.g., FTK and Encase, or are limited in application scope. For instance, E-mail Mining Toolkit and Paraben e-mail examiner do not address the issue of anonymity.

The problem of criminal information mining is, to design an approach to *automatically* perform a *semantic analysis* of textual content (usually large archives) of online documents for collecting forensically relevant information. The extracted information needs to be precise, creditable, and interpretable with a certain degree of acceptance. The expert witness needs to present information in different levels of granularity to enhance interpretability and intuitiveness.

1.2 Objectives

The main objective of this research is to develop a data mining framework for forensic analysis of online documents by:

- extracting patterns of authorial attributes to address three problems of authorship analysis–authorship attribution, authorship characterization, and authorship verification;
- mining criminal data to extract knowledge relevant to cybercrime investigation; and
- supporting the findings in terms of interpretability, intuitiveness, and preciseness.

1.3 Contributions

We have developed a set of methods to pursue our objectives and to fill the research gap identified in the above mentioned problem scenarios. The contributions are summarized under the following two main headings: authorship analysis and criminal information mining.

Authorship Analysis

To overcome the limitations of existing authorship techniques, in this study we introduce a novel approach of authorship analysis in which the author-specific writeprint is extracted. To concisely model the writeprint of an individual we borrow the concept of *frequent pattern* [7] from data mining to capture the combinations of features that frequently occur in an individual's online documents. Frequent pattern mining has been proven to be a very successful data mining technique for finding hidden patterns in DNA sequences, customer purchasing habits, security intrusions, and has been used in many other applications of pattern recognition.

The extracted writeprint is applicable to most of the authorship analysis problems

discussed in this thesis including authorship identification, characterization, and verification. Similarly, our method can be employed on all kinds of online documents, e.g., e-mails and chat logs. The extracted writeprint is easy to interpret and understand as it is simply the combination of stylometric features. It would be hard for an accused person to rebut or deny charges because the findings can be traced in his/her sample documents. Following are some of the major contributions of our proposed authorship approach.

- *Frequent pattern-based writeprint:* We precisely model the writeprint of a suspect by employing the concept of *frequent patterns* [60]. Intuitively, the writeprint of a suspect is the combination of stylistic features that are frequent in her text messages but not in other suspects' messages. To ensure the uniqueness of the writeprint among the suspects, our approach ensures that any two writeprints among suspects are disjoint, meaning they do not share any frequent pattern. This is the first work that presents a data mining solution based on the frequent pattern-based writeprint to address all three authorship analysis problems discussed in Section 1.1.1.

- *Capturing stylistic variation:* Our insight is that a person may have multiple writing styles depending on the recipients and the context of a message. We present an algorithm to precisely model the sub-writeprints of a suspect using the concept of frequent patterns. Experimental results suggest that the identification of sub-writeprints can improve the accuracy of authorship analysis. Most importantly, the sub-writeprint reveals the fine-grained writing styles of an individual, which can be valuable information for investigators or authorship analysis experts [50].

- *Analysis based on different training sample sizes:* Traditional authorship analysis methods often require a reasonably large volume of training samples in order to build a classification model. Our proposed method is effective even if only a few training samples exist. In case, no training sample is available, our approach can infer the characteristics of the authors based on the stylometric features in the anonymous text messages.

- *Presentable evidence:* A writeprint is a combination of stylometric features that are frequently found in a suspect's text messages. Given that the concept is easy to understand, an investigator can present the writeprint and explain the finding in a court of law. Some traditional authorship identification methods, such as SVM and neural networks [104, 121], do not share the same merit.

- *Remove burden from investigator:* One question frequently raised by cybercrime investigators is how to determine the right set of stylometric features that should be used for the authorship analysis case in hand. Adding unrelated stylometric features can distort the accuracy of an analysis. Our notion of frequent pattern-based writeprint resolves the problem because insignificant patterns are not frequent and, therefore, do not appear in the writeprint. Thus, an investigator can simply add all available stylometric features without worrying about degrading the quality.

- *Stylometry-based clustering:* Content-based clustering for dividing documents into different groups has long been used. Our experimental results suggest that clustering by stylometric features is a promising technique to group online messages

written by the same person into one cluster. The notion of stylometry-based clustering is applicable in most authorship analysis problems [58].

Criminal Information Mining

The contributions of our criminal information mining module are given below.

- *Analyzing unstructured data:* Most criminal information mining studies focus on structured documents, e.g., police narratives [25]; our data mining framework is designed for analyzing unstructured data, e.g., chat logs. Structured documents are easy to analyze as they are large in size, formal in style and composition, and properly compiled following common syntactic and grammatical rules, as compared to online messages, which are usually written in 'para' language.

- *Topic identification without training data:* The traditional topic identification techniques generally determine the topic of a given document from a list of some predefined topic categories. For this, the investigator is assumed to have sample documents for each category to train a classifier. Our approach does not require any training data and can dynamically assign topic to a new document based solely on its content.

- *Semantic analysis:* To effectively analyze the text discourse, we use the word similarity as well as the relatedness measure, defined in WordNet in word clustering and topic identification steps of our method. Our approach can disambiguate whether a word is used in its normal meaning or in its malicious meaning.

- *Adapting expert knowledge to the data mining process:* A cybercrime investigator

can employ a taxonomy of the street terms used for different crimes in our presented approach to guide the analysis process. The taxonomy can be extracted from large collections of criminal conversation.

1.4 Organization of Thesis

The remainder of the thesis is organized as follows. Chapter 2 gives an overview of the current literature on the subjects that are related to the problems addressed in this thesis. The literature review consists of two parts: authorship analysis and criminal information mining. Chapter 3 describes the analysis techniques employed on the message header. Chapter 4 proposes a novel approach of frequent pattern-based writeprint extraction for addressing the problem of authorship attribution. We extend the approach to address the attribution problem in the presence of stylistic variation. Chapter 5 defines a new scenario of authorship identification in which very few training samples are available.

Chapter 6 studies the authorship characterization problem and proposes a technique to infer the sociolinguistic attributes of the potential author of a given anonymous message. Chapter 7 defines the authorship verification problem and proposes a method based on the NIST speaker-recognition evaluation framework [72]. Chapter 8 discusses a criminal information mining approach for analyzing the textual content of online messages. Chapter 9 concludes the thesis and identifies directions for future research.

Chapter 2

Related Work

In this chapter, we present a review of state-of-the-art techniques developed in the areas of authorship analysis and criminal information mining. Authorship is studied in terms of stylometric features and analysis techniques. The analysis techniques are further divided into three groups for addressing the three subproblems, i.e., authorship attribution, authorship characterization, and authorship verification. In the literature, criminal information mining is studied under the topics of named entity recognition, link mining, text summarization, and concept mining.

In the current study we provide a review of the main approaches proposed in each of the aforesaid research areas, along with their shortfalls. To overcome the identified shortfalls, we briefly discuss our proposed solution.

This chapter is organized as follows: In Section 2.1, we discuss the special characteristics of online communication documents. In Section 2.2, we give a brief description of a data mining benchmark toolkit, Waikato Environment for Knowledge Analysis

(WEKA) [111], used in many existing authorship classification studies. In Section 2.3, we give a review of the commonly used stylometric features, authorship attribution techniques, authorship characterization methods, and authorship verification approaches. Section 2.4 reviews the different language processing and text mining techniques developed for discovering criminal information. We conclude the chapter in Section 2.5.

2.1 Characteristics of Online Messages

Online documents or electronic discourses are written communications exchanged between people over the Internet. The mode of communication of online documents can be synchronous, such as chat logs, or asynchronous, such as e-mail messages and web forums [4]. Authorship analysis of online documents is more challenging than analyzing traditional documents due to their special characteristics of size and composition [34]. According to [42], "Electronic discourse or online document is neither here nor there, neither pure writing nor pure speech but somewhere in between."

The traditional literary works such as books and essays are rich sources of learning about the writing style of their authors. Because literary works are usually large in size ranging from few paragraphs to several hundred pages. They are generally well-structured in composition following definite syntactic and grammatical rules. Most traditional documents are written in formal way and are intended for a variety of readers. Moreover, the availability of natural language-processing tools and techniques make it easy to improve the quality of these documents by removing spelling and idiosyncratic mistakes. The study of stylometric features has long been very successful in resolving ownership

disputes over literary and conventional writings [74].

Online documents, on the other hand, are short in size, varying from a few words to a few paragraphs, and often they do not follow definite syntactic and/or grammatical rules. Therefore, it is hard to learn about the writing habits of their authors from such documents. Ledger and Merriam [68], for instance, have established that authorship analysis results would not be significant for texts containing fewer than 500 words. Moreover, online documents are interactive, informal in style, and are usually written in 'para' language. People usually do not pay attention to their spelling and grammatical mistakes. Therefore, the analytical techniques that are successful in addressing authorship issues over literary and historic works may not produce trustable results in the context of online document analysis.

Electronic discourses such as e-mail documents do have certain properties that help researchers compare individuals' writing styles. One can find more e-mail documents for analysis; every e-mail user writes, on the average, 6-10 e-mails per day. Similarly, additional information contained in the header (e.g., time stamps), subject line, and/or attachment(s), are helpful in learning about the writing style of a user. Moreover, e-mails are rich in structural features, e.g., greetings, general layout, and the sender's contact information, that are powerful discriminators of writing styles [34].

2.2 WEKA

Waikato Environment for Knowledge Analysis (WEKA) is a collection of state-of-the-art machine learning algorithms and data processing tools used for solving data mining

problems. WEKA has been developed at the University of Waikato in New Zealand. It is written in Java and distributed under the terms of a general public license. Most of the WEKA functionality can be used both from within the WEKA toolkit and outside the toolkit, i.e., they can be called from a Java program.

WEKA provides extensive support for the whole process of data mining including preparing data, constructing and evaluating learning algorithms, and visualizing the input data, including results of the learning process. The WEKA includes methods for most standard data mining problems: regression, classification, clustering, association rule mining, and attribute selection.

Classification methods implemented in WEKA [111], namely Ensemble of Nested Dichotomies (END) [44], J48 [87], Radial Basis Function Network (RBFNetwork) [18], NaiveBayes [91], and BayesNet [82] are commonly used for authorship analysis. The decision tree classifier C4.5 implemented in WEKA is denoted as J48. The three widely used clustering algorithms, Expectation-Maximization (EM), *k*-means, and bisecting *k*-means, are implemented in WEKA.

The WEKA native data file is the Attribute-Relation File Format (ARFF). It is an ASCII text file that describes a list of instances sharing a set of attributes. A sample ARFF consists of two sections: the *header* and the *data*, as shown in Figure 2.1. The header, called the data declaration section, contains names of attributes followed by their type. The type of an attribute can be numeric (integer or real), nominal, string, or date, as depicted in Figure 2.1.

The data section starts with the reserved word data preceded by the symbol '@' and

is followed by rows of attribute values. The attributes are ordered having a one-to-one association with the attributes defined in the declaration section. Each row represents one instance of the declared attributes. The missing values are denoted by a question mark within the respective position in the row. Values of string and nominal attributes are case sensitive.

```
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute run {yes, no}

@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no
```

Figure 2.1: A sample ARFF file

2.3 Authorship Analysis

Authorship analysis is the study of linguistic and computational characteristics of written documents of individuals [13, 20]. Writing styles or specific writing traits extracted from authors' previously written documents can be used to differentiate one person from another [77]. Writing styles are studied in terms of mainly four types of stylometric features:

lexical, syntactic, structural, and content-specific. Analytical authorship techniques employed so far include univariate and multivariate statistics [21, 43], machine learning processes such as support vector machine and decision trees [33, 121], and frequent-patterns mining [60].

Most previous contributions on authorship attribution are applications of text classification techniques [33]. The process starts by identifying a set of a person's writing style features of that are relatively common in most of her works. A classifier is trained on the collected writing style features to build a model, which is then used to identify the most plausible author of anonymous documents.

In the literature, the authorship problem is generally studied from the following three perspectives [34, 60].

- *Authorship attribution or identification* is applied to an anonymous document to determine the likelihood of a specific author by examining his previously known documents.

- *Authorship profiling or characterization* is used to characterize authors based on their background and demographic information including gender, education level, and linguistic and cultural attachment.

- *Similarity detection* or *authorship verification* is used to detect plagiarism, copyright violation, or intellectual property theft. It is applied to determine whether or not any given two pieces of anonymous work, e.g., textual online document, program code, or algorithm, are produced by the same entity [49].

The related work of the aforementioned three aspects of authorship analysis is preceded by a literature review of the stylometric features employed in authorship studies.

2.3.1 Stylometric Features

In traditional criminal investigation cases fingerprints are used to uniquely identify criminals. In the present era of the computer and World Wide Web, the nature of most crimes and the tools used to commit crimes have changed. Traditional tools and techniques may no longer be applicable in prosecuting cybercriminals in a court of law. The statistical study of stylometric features, called stylometry, shows that individuals can be identified by their relatively consistent writing styles. The writing style of an individual is defined in terms of word usage, selection of special characters, composition of sentences and paragraphs, and organization of sentences into paragraphs and paragraphs into documents.

Rudman has identified more than 1000 stylometric features in his study [92]. But there is no such feature set that is optimized and equally applicable to all people and in all domains. However, previous authorship studies [13,20,34,122] contain lexical, syntactic, structural, and content-specific features. Other features studied in authorship literature include idiosyncracies [4], n-grams (e.g., bigrams and trigrams), and frequency of part-of-speech tags [11]. Brief description and the relative discriminating capability of the main feature types are given below.

- *Lexical features* are used to learn about an individual's preferred use of isolated characters and words. These include frequency of individual letters of alphabets (26 letters of English), total number of upper case letters, capital letters used in the

beginning of sentences, average number of characters per word, and average number of characters per sentence. The use of such features indicates an individual's preference for certain special characters or symbols or the preferred choice of selecting certain units. For instance, some people prefer to use the '$' symbol instead of the word 'dollar', '%' for 'percent', and '#' instead of writing the word 'number.'

Word-based lexical features, including word length distribution, words per sentence, and vocabulary richness, were very effective in earlier authorship studies [56, 116,117]. Recent studies on e-mail authorship analysis [34,121] indicate that word-based stylometry such as vocabulary richness is not very effective for two reasons. First, e-mail messages and online documents are very short compared to literary and poetry works. Second, word-oriented features are mostly context dependent and can be consciously controlled by people.

- *Syntactic features* include content-independent all-purpose words, e.g., 'though', 'where', and 'your'; punctuation, e.g., '!' and ':'; and part-of-speech tags. Mosteller and Wallace [78] were the first to show the effectiveness of the function words in addressing the issue of Federalist Papers [116]. Burrows [20] used 30-50 typical function words for authorship attribution. Subsequent studies [13] have validated the discriminating power of punctuation and function words. Zheng et al. [121] used more than 150 function words. Stamatatos et al. [100] used frequencies of part-of-speech tags, passive account, and nominalization count for authorship analysis and document genre identification.

- *Structural features* are helpful to learn how an individual organizes the layout and

structure of his/her documents. For instance, how are sentences organized within paragraphs, and paragraphs within documents? Structural features were first suggested by de Vel et al. [29, 34] for e-mail authorship attribution. In addition to the general structural features, they used features specific to e-mails such as the presence/absence of greetings and farewell remarks and their position within the e-mail body. Moreover, within e-mails some people use first/last name as a signature while others prefer to include their job title and mailing address as well. Malicious e-mails contain no signatures and in some cases may contain fake signatures.

- *Content-specific features* are used to characterize certain activities, discussion forums, or interest groups by a few keywords or terms. For instance, people involved in cybercrimes (spamming, phishing, and intellectual property theft) commonly use (street words) 'sexy', 'snow', 'download', 'click here', and 'safe', etc. Usually term taxonomy built for one domain is not applicable in other domain and can even vary from person to person in the same domain. Zheng et al. [121] used around 11 keywords (such as 'sexy', 'for sale', and 'obo') from the cybercrime taxonomy in authorship analysis experimentations. A more comprehensive list of stylistic features including idiosyncratic features was used in [4].

- *Idiosyncratic features* include common spelling mistakes, e.g., transcribing 'f' instead of 'ph' (as in the word phishing) and grammatical mistakes, e.g., writing sentences with the incorrect form of verbs. The list of such characteristics varies from person to person and is difficult to control. Gamon [47] achieved high accuracy by combining certain features including part-of-speech trigrams, function

word frequencies, and features derived from semantic graphs.

2.3.2 Authorship Attribution

The problem of authorship attribution or authorship identification in the context of online documents is to identify the true author of a disputed anonymous document. In forensic science an individual can be uniquely identified by his/her fingerprint. Likewise, in cyber forensics, an investigator would like to identify the specific writing styles, called wordprint or writeprint, of potential suspects and then use them to develop a model. The writeprint of a suspect is extracted from her previously written documents. The model is applied to the disputed document to identify its true author among the suspects. In forensic analysis the investigator is required to support her findings by convincing arguments in a court of law.

In the literature, authorship identification is considered as a text categorization or text classification problem. The process starts by data cleaning followed by feature extraction and normalization. Each document of a suspect is converted into a feature vector using vector space model representation [94]; the suspect represents the class label. The feature values are calculated by using the five commonly used stylometric features discussed in Chapter 4. The extracted features are bifurcated into two groups, training and testing sets. The training set is used to develop a classification model while the testing set is used to validate the developed model by assuming the class labels are not known. Common classifiers include decision tree [87], neural networks [70], and support vector machine [61].

If the error approximation is below a certain acceptable threshold, the model is employed. The disputed anonymous document is preprocessed and converted into a feature vector in a manner similar to the one adopted for known documents. Using the developed model, the conceivable class label of the unseen document is identified. The class label indicates the author of the document in question. Usually, the larger the training set the better the accuracy of the model. The accuracy of the model is gauged by employing the popular functions called *precision* and *recall*, described in [78].

The difference between traditional text classification and authorship classification is that in text categorization syntactic features, e.g., punctuation, all-purpose stop words, and spaces, are dropped and the features list includes topic-dependent words, while in authorship problems, topic words or content-dependent words are removed and the features are calculated in terms of style markers or syntactic features. Similarly, in text categorization problems the class label is the topic title among the predefined document categories, while in authorship attribution the class label is the author of the document.

Most authorship attribution studies differ in terms of the stylometric features used and the type of classifiers employed. For instance, Teng et al. [104] and de Vel [33] applied SVM classification model over a set of stylistic and structural features for e-mail authorship attribution. de Vel et al. [34] and Corney et al. [29] applied SVM on an e-mail dataset and discovered the usefulness of structural features for e-mail authorship attribution. They have also studied the effects of varying the number of authors and sample size on the attribution accuracy.

Zheng et al. [121, 122] and Li et al. [69] used a comprehensive set of lexical, syntactic, and structural features including 10-11 content-specific keywords. They used three classifiers including C4.5, neural networks, and SVM for authorship identification of online documents written in English and Chinese languages. Van Halteren [107] used a set of linguistic features for authorship attribution of students essays. In [65], different classifiers were evaluated for authorship identification of chat logs. Zhao and Zobel [120] have studied the effectiveness of function words in authorship problems by applying different classifiers.

de Vel [34] found that by increasing the number of function words from 122 to 320, the performance of SVM drops, due to the scalability problem of SVM. This result also illustrates that adding more features does not necessarily improve accuracy. In contrast, the focus of this thesis is to identify the combinations of key features that can differentiate the writing styles of different suspects and filter out the useless features that do not contribute towards authorship identification.

Some research proposals [34, 35] have recognized the contextual and temporal change in the writing style of a person, although most choose to ignore such variations and focus on obtaining the permanent writing traits of an individual. Therefore, they extract stylometric features from the entire sample dataset of a suspect, disregarding the context and the type of recipient of a message. In fact, the writing style of an individual varies from recipient to recipient and evolves with the passage of time and with the context in which a message is written [34].

Style variation is a factor of the commonly used four types of writing style features.

For example, the change in the topic of an online message is indicated by the relative composition of words and phrases. Official messages may contain more formal words and phrases that may result in an increased value of vocabulary richness. Similarly, syntactical features, including punctuation, hyphenation, and distribution of function words, are usually more frequent in online text written to the top management of a company.

Moreover, the ratio of spelling and grammatical mistakes is usually higher in electronic discourse sent to a friend than to a co-worker. More specifically malicious e-mails may not contain the signatures and contact information. Instead, malicious messages may contain more fancy and charming words that are appealing and attractive to the target victims. Words like 'congratulations!', 'hurry up', 'free download' and 'obo' are commonly found in spamming messages.

Similarly, the content and writing styles found in illegitimate messages are overshadowed by regular messages as the malicious messages are usually much fewer in number than regular messages. The analytical techniques employed over such intermingled writing samples would produce misleading results. In the current study we propose techniques for capturing the stylistic variation of a suspect to improve the attribution accuracy.

2.3.3 Authorship Characterization

Authorship characterization [29, 64] is applied to collect sociolinguistic attributes such as gender, age, occupation, and educational level, of the potential author of an anonymous document. In the literature, authorship characterization is addressed as a text classification problem. Generally, a classification model is developed by using the textual

documents previously written by the sample population. The developed model is applied to the anonymous document to infer the sociolinguistic characteristics of the potential anonymous author.

Corney et al. [29], Koppel et al. [63, 64], and Argamon et al. [12] studied the effects of gender-preferential attributes on authorship analysis. Other profiling studies have discussed educational level [29], age, language background [64], and so on. To address the same issue in the context of chat dataset, some techniques have been proposed in [65] for predicting the potential author of a chat conversation. The proposed technique is employed to collect sociolinguistic and demographic information such as gender, age, and occupation of the writer of an anonymous chat segment.

Abbasi and Chen [5] applied similarity detection techniques on customer feedback to identify fake entities in the online marketplace. In [29, 64], authorship profiling was applied to collect demographic and sociolinguistic attributes of the potential author of a disputed document.

Existing characterization studies vary in terms of type of classifiers used, dimension of characteristics inferred, and nature of documents analyzed. For instance, Corney et al. [29] and de Vel et al. [36] used support vector machine to infer the gender, educational level, and language background of an e-mail dataset. Koppel et al. [64] applied Bayesian regression function to predict the gender, age, and native language of the perceived author of anonymous text.

Most characterization studies are based on classifiers, which are not suited for forensic analysis due to some limitations, discussed in Section 2.3.5. Our method is

founded on frequent pattern-based writeprint extraction, representing the unique writing style of an individual. Unlike traditional techniques, our method does not require large training data for producing trustable results. Similarly, the proposed approach can be applied to most text discourses, although the current study is focused on a blog dataset. The class dimensions of the authors include gender and region.

2.3.4 Authorship Verification

Unlike authorship attribution and authorship characterization, where the problem is clearly defined, there is no consensus on how to precisely define the problem in authorship verification studies. Some studies, e.g., [4, 33], have considered it as a similarity detection problem: to determine whether two given objects are produced by the same entity or not, without knowing the actual entity.

Internet-based reputation systems, used in online markets, are manipulated by using multiple aliases of the same individual. Novak et al. [79] proposed a new algorithm to identify when two aliases belong to the same individual, while preserving privacy. The technique has been successfully applied to postings of different bulletin boards, achieving more than 90% accuracy. To address the same issue of similarity detection, Abbasi and Chen [4,5] proposed a novel technique called *writeprints* for authorship identification and similarity detection. They used an extended feature list including idiosyncratic features in their experimentations. In similarity detection, they took an anonymous entity, compared it with all other entities, and then calculated a score. If the score is above a certain predefined value, the entity in hand is clustered with the matched entity.

Following the same notion of verification, Halteren [107] proposed a relatively different approach called linguistic profiling. In this study he proposed some distance and scoring functions for creating profiles for a group of example data. The average feature counts for each author was compared with a general stylistic profile built from the training samples of widely selected authors. The study focused on detecting similarity between student essays for plagiarism and identity theft.

The more common notion of authorship verification is to confirm whether or not the suspect is the author of a disputed anonymous text. Some studies address authorship verification as a one-class classification problem (e.g., [120] and [71]) while others (e.g., [63] and [64]) as a two-class text classification problem. For instance, Manevitz et al. [71] investigated the problem as follows: Given a disputed document together with sample documents of the potential suspect, the task is to verify whether or not a given document is written by the suspect in question. Documents written by sample population are labeled as 'outlier' in their study. A classification model is developed using the stylometric features extracted from the collected documents. The built model is applied to identify the class label of the given anonymous document.

A slightly modified version of the one-class approach called 'imposter' is the two-class problem proposed by Koppel et al. [63]. According to this study, the known documents of the potential suspect are labeled as 'S' and that of the sample population as 'imposter'. A classification model is developed using the stylometric features extracted from these documents. The anonymous document is divided into different chunks and each chunk is given to the model to predict its class. The method fails to work if the

documents of the 'imposter' and the suspect are closely similar.

An opposite approach would be to train one model for *S* and for *not-S* and then employ a trained model to determine the degree of dissimilarity between them [64]. In this study the authors employed the traditional 10-fold cross-validation approach. If the validation accuracy is high, it is concluded that *S* did not write the document in question. Otherwise the model fails to assign a class label.

A relatively new approach, called 'unmasking' [64], is the extension of the 'imposter' method. In this study the authors attempted to quantify the dissimilarity between the documents of the suspect and that of the 'imposter.' The experimental results reported indicate that for achieving trustable results the method is suitable in situations where the document in question is at least 5000 words long. This is nearly impossible in the case of online documents.

In this thesis we address authorship verification as a two-class classification problem. We develop a universal background model (UBM) by using documents from a large population. We borrow the techniques from the SRE framework [72] to train and validate the representative model. The SRE framework is very successful in the speaker recognition community. Similarly, evaluation measures such as DCF, minDCF, and EER, used in the aforesaid framework, are suited for forensic studies.

2.3.5 Limitations of existing Authorship Techniques

Most of the existing authorship analysis techniques are primarily based on some commonly used classifiers. These classifiers can be broadly divided into three main categories: probabilistic [91], decision trees [88, 89], and support vector machine [30]. Each of these classifiers has its own limitations in terms of classification accuracy, scalability, and interpretability. Probabilistic Naive Bayes classifiers and batch linear classifiers (Rocchio) seem to be the worst of the learning-based classifiers, while SVM appears to be the best in terms of classification accuracy [96].

Similarly, while building a decision tree a decision node is constructed by simply considering the local information of one attribute; therefore, it fails to capture the combined effect of several features. In contrast, SVM avoids such a problem by considering all features when a hyperplane is created. However, SVM is like a black-box function that takes some input, i.e., a malicious message, and provides an output, i.e., the author. It fails to provide an intuitive explanation of how it arrives at a certain conclusion. Therefore, SVM may not be the best choice in the context of forensic investigation, where collecting credible evidence is one of the primary objectives.

Most classifiers would require sufficiently large training data to produce acceptable classification accuracies. The collected training samples from the suspects in criminal investigation cases are not always enough to train a classifier. Therefore, the need is to design an approach that can work even with small training data. Similarly, most authorship techniques that are successful in resolving authorial disputes of structured documents, e.g., books and formal reports, may not produce trustable results in the context of online

messages due to their short size and casual content.

To overcome the limitations of existing authorship techniques, we develop a novel approach of authorship analysis. In this method, we create a unique *writeprint* for each suspect based on her previously written documents. The concept of writeprint is based on the idea of *frequent pattern* [7], a data mining technique. Frequent-pattern mining has been very successful in finding interesting patterns in large archives of documents analyzed for identification of customer purchasing habits, cataloguing objects in large super stores, intrusion detection systems, and traffic classification.

2.4 Criminal Information Mining

The textual content of a document can be analyzed to collect forensically relevant information that can be used to answer the following questions: Who is the potential author of a text discourse? What are the pertinent suspicious entities mentioned within a document? Are these entities related to each other? What concepts and topics are discussed in the document(s)? [12]. Predictive machine learning measures and natural language processing techniques are applied to extract information. Authorship analysis techniques are used to learn about the potential author of an anonymous discourse [60]. Social networking [24] and link analysis techniques [95] are applied to identify covert association between crime entities. Similarly, topic identification or topic detection is employed to identify the topic or genre of a document [9]. Text summarization methods [14,15,32,109] are applied to extract the summary of a potentially large collection of documents.

Detailed description of the aforementioned areas is given in the following paragraphs.

Zheng et al. [121, 122] developed an authorship analysis framework for identifying the true author of anonymous online documents. They built a classification model based on the previously written documents of potential suspects, and then employed the model to identify the true author of a given disputed document. Using a similar approach, in [65] the authors proposed authorship attribution techniques for chat dataset. In [29,64], authorship profiling was applied to text documents to collect demographic and sociolinguistic attributes (e.g., gender, age, and occupation) of the potential author of a disputed document. Abbasi and Chen [5] applied similarity detection techniques on customer feedback to identify fake entities in an online marketplace. In most of these studies the classification models used are: (1) probabilistic classifiers (e.g., Bayesian classifiers [91] and its variants), (2) decision trees [87], and (3) support vector machine (SVM) [61] and its variants.

Named Entity Recognition (NER), a branch of natural language processing, is used to identify information associated with an entity, such as the name of a person, place, or company; contact information such as phone, e-mail, or URL; or other attributes such as date-of-birth, vehicle number, or assurance number [26]. Chen et al. [24] employed named entity recognition techniques for extracting criminal identities from police narratives and other suspicious online documents. Minkov et al. [76] proposed techniques for extracting a named entity from informal documents including e-mail messages. Sometimes cybercriminals use identity deception tactics to falsify their true identities. Wang

et al. [108] proposed an adaptive detection algorithm for detecting masqueraded criminal identities. Carvalho and Cohen [22] studied techniques for identifying user signatures and the 'reply part' from the e-mail body.

To facilitate crime investigation process, Chau et al. [95] applied new link analysis techniques to the Tucson police department database to identify covert association between crime entities. The proposed techniques, including shortest path algorithm, co-occurrence analysis, and a heuristic approach, have been successful in identifying associations and determining their importance. The study [23] applied association rules mining techniques to suspicious web sites, called dark web, for identifying online communication between those accused of the 9/11 attacks.

Topic identification, within a corpus of text documents, is the extraction of pertinent content related to a known topic or the topic to be listed [9]. In the literature of information retrieval and browsing, topic identification is generally addressed either in a supervised way or an unsupervised way [85]. In the *supervised* way, the problem of topic discovery is handled as a text classification or text categorization problem [96]. According to this approach, usually there exit some predefined topic categories with example documents for each category. To infer the topic of an unknown document, a classification model is developed on the given sample documents. Similarly, unsupervised learning or clustering is applied to identify the pertinent groups of objects based on some similarity measure.

Pendar [83] has applied automatic text categorization techniques on suspicious chat conversation to identify online sexual predators. Each given chat session is converted into a vector of attributes using bag-of-words model. Attributes are the frequencies of

word unigrams, bigrams, and trigrams. The words that appear either very rarely (say once) or very frequently (say above 95%) in a given chat log are deleted. They develop a classification model by applying SVM and k-NN classifiers on some previously known predators' chat conversations. The developed model is then employed to identify the predator (or pseudo-predator) communication from a teenager (i.e., a victim) communication. Elnahrawy [40] compared the performance of three classifiers, i.e., Naive Bayes, SVM, and K-nearest neighbor, for automatically monitoring chat conversation following the general text categorization approach. Studies [39, 62, 80] focused on topic identification of chat logs from a list of some predefined topics. Zhang et al. [119] have developed text classification techniques for automatic key phrase extraction in Web documents.

The *unsupervised* topic identification or topic discovery is achieved by applying content-based clustering. *Clustering* is used to uncover useful and interesting text patterns in a corpus without knowing any background knowledge [85]. Once each document is converted into term vector and the pairwise distance between the term vectors is defined, a clustering algorithm is applied to divide the documents into groups. The documents of a cluster are similar together and are dissimilar from documents of other clusters. Once the documents are clustered, each cluster is labeled with the topic words. The topic words or the cluster label is identified by using different techniques. The simplest way is to identify the words that are found frequently with a particular cluster. There are two main categories of clustering algorithms: partitioned or hierarchial. In hierarchial clustering, documents are diagramed into a tree-like structure called a dendrogram [31]. Topics at the top level are more general, becoming more specific while descending toward the terminal

nodes. The documents associated to each topic are linked to that node.

In [113], the specific attributes of chat users and the relation between users within a chat room are visually displayed. The authors used metaphors for creating visual data *portraits* of the attributes extracted from chat content and the patterns of conversation of users. Example of attributes are: time since initial posting, participation frequency of a user in a chat room or in a topic, and number of responses to a posting. Bingham et al. [16] developed a chat analysis tool, called ChatTrack, for summarizing and filtering chat logs. A classifier is trained on a set of predefined concepts or topics with sample documents. The classifier then creates a vector of high frequency words for each topic category. Next, a conceptual profile is created for a selected chat conversation or chat user by training a classifier on the selected chat sessions. The trained classifier is used to create a vector of selected words. Finally, using the cosine similarity measure [94], the similarity between the profile vector and the predefined concept vectors is calculated. There are more than 1565 predefined concepts' hierarchies and their sample documents.

A criminal information mining framework, proposed in [25], was designed by integrating state-of-the-art data mining techniques such as link analysis, association rule mining, and social network analysis. The developed framework is believed to have the capability of identifying different kinds of crimes. The main focus of the framework is to collect network level information (i.e., web addresses). The framework can analyze only structured documents such as police narratives. Xiang et al. [112] focused on visualizing crime data for facilitating the work of an investigator.

In order to automatically analyze large archives of online documents, an investigator requires an integrated software tool. In the current study we employ most of the aforesaid text mining techniques to design and implement a framework in order to help an investigator perform a multi-stage analysis of electronic discourse including chat logs. The framework takes suspicious chat logs as input, extracts named entities, divides them into different groups, and then retrieves chat logs of each group for further processing. We extract keywords and summary from each chat collection, which are then processed to extract *concepts* and *key concepts* representing the topic of the chat log in question. The extracted suspicious groups and their relationships are visualized in more intuitive fashion. The state-of-the-art techniques employed to accomplish the abovementioned tasks are discussed below.

We employ the widely used Stanford Named Entity Recognizer, called CRFClassifier [1] to extract the named entities. The tool is tested on popular corpora such as MUC-6, MUC-7, and ACE. To identify the relationships between the entities for determining cliques, we apply *frequent patterns mining* techniques. Next, we use two criteria to extract the keywords: first, the word matches with the street term(s) listed in the domain-specific cybercrimes taxonomy; second, the frequency of the word is above the user-defined threshold. The sentences in which one or more keywords appear constitute the summary.

The extracted keywords are converted into concepts and the concepts are converted into key concepts and topics by using WordNet. The WordNet is a lexical database, described in Chapter 8. The selection of WordNet for the purpose of concept mining and

[1] http://nlp.stanford.edu/software/CRF-NER.shtml

topic mining is based on: (1) words are organized into hierarchies of concepts called synset (synonyms sets); (2) the hierarchies are based on both *similarity* and *relatedness*; (3) Hyponymy, which means that the WordNet synsets are structured in such a way that abstract concepts (called hypernyms) are given at a higher level while more specific concepts (called hyponyms) are given at a lower level; and (4) a computer-readable database of commonly used words.

2.5 Summary

In this chapter, we have presented state-of-the-art techniques developed in the areas of authorship analysis, stylometric features, and criminal information mining. In the authorship domain, we focus on the common classifiers used in different authorship analysis studies. Stylometric features used in most authorship studies fall into five main categories including lexical, syntactic, structural, content-specific, and idiosyncratic. The literature of criminal information mining broadly covers the research areas of natural language processing, information retrieving, link analysis, and social network analysis.

Chapter 3

Header-level Investigation

In this chapter we provide a brief description of the methods we employ for collecting initial information about a given suspicious dataset. The header content is usually the immediate source for collecting preliminary information about a given collection of suspicious online messages. The statistical analysis of an e-mail corpus–identifying all the senders, the recipients associated with each sender, and the frequency of messages exchanged between users–helps an investigator understand the overall picture. The structure of a person's social network, extracted from a dataset, manifests information about his/her behavior with other people, including her friends, colleagues, and family members. In some investigations it is important to identify the physical location of the users. This can be achieved by applying geographical localization and map retrieval techniques on the e-mail addresses. Moreover, classifying messages into predefined topics can be achieved by applying traditional text categorization techniques.

The remainder of the chapter is organized as follows: Section 3.1 calculates simple

statistics on a given message collection. Section 3.2 summarizes the importance of social networking techniques for learning about the general behavior of the users. Section 3.3 describes map retrieval techniques used for mapping an e-mail address to its physical location. Section 3.4 describes the application of text classification and clustering techniques to message analysis. Section 3.5 concludes the chapter.

3.1 Statistical Analysis

A statistical analysis of a message dataset analyzing the flow of messages between users is important during the early stages of investigation. For instance, identifying the total number of users (i.e., senders/recipients) and the distribution of messages per sender-domain and per recipient-domain gives an overview of the entire message collection, as shown in Figure 3.1. Similarly, the mailing frequency during different parts of the day and night and the average response time of users are calculated to model their behavior. For instance, an e-mail user may send more messages to her co-workers during the day rather than night. Similarly, calculating the average size of a message and its attachment (if one exists) and identifying the format of the message attachment are helpful in creating a user's profile. The user profile is used in anomaly detection systems for identifying the abnormal behavior of users.

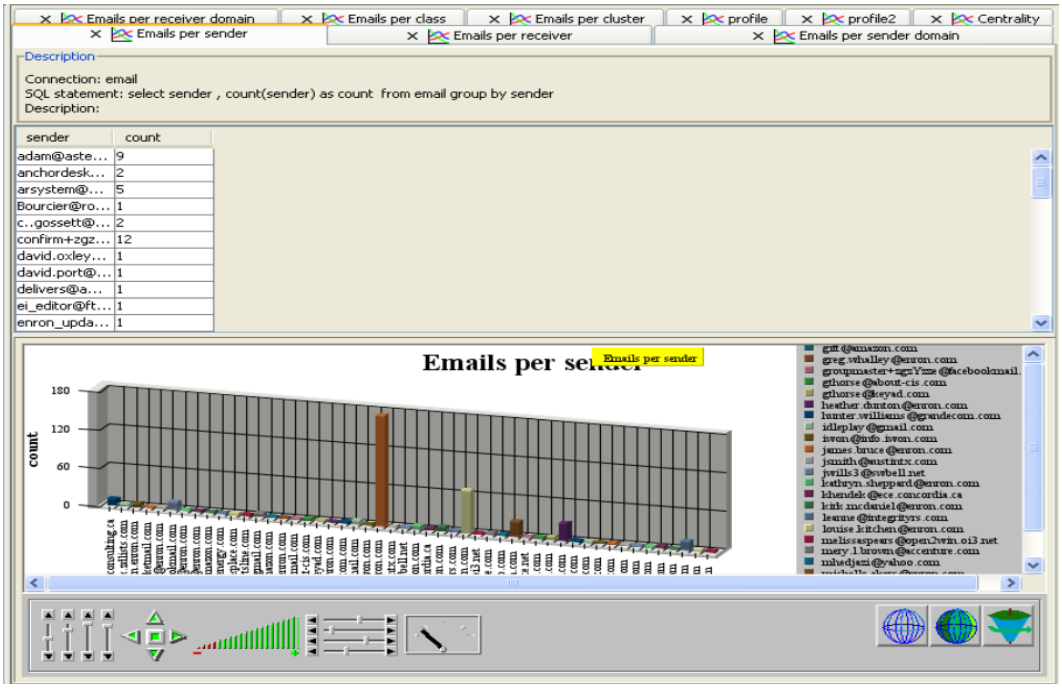


Figure 3.1: Statistics calculated for an e-mail dataset

3.2 Social Network Analysis

Social network analysis is the study of analyzing communication links between people. The social network for an e-mail dataset can be depicted as a graph, where the nodes represent the senders and recipients, and the edges represent the flow of e-mail messages between them. The structure of a user's social network, extracted from his/her e-mails manifests a great deal of information about his/her behavior within the community of friends, colleagues, and family members. This information can be used to answer the following questions [17]: for example, (1) How often does a person maintain a relationship with a group of people, and for how long? (2) Do these people have regular interactions and can these interactions be distinguished based on roles such as work, friendship, and family? (3) What type of views are a particular group of people exchanging? For instance,

the analysis of a criminal network can be used to discover interesting information about potential suspects and periods of their suspicious activities. In this chapter, we do not analyze the message body but rather focus on the message header.

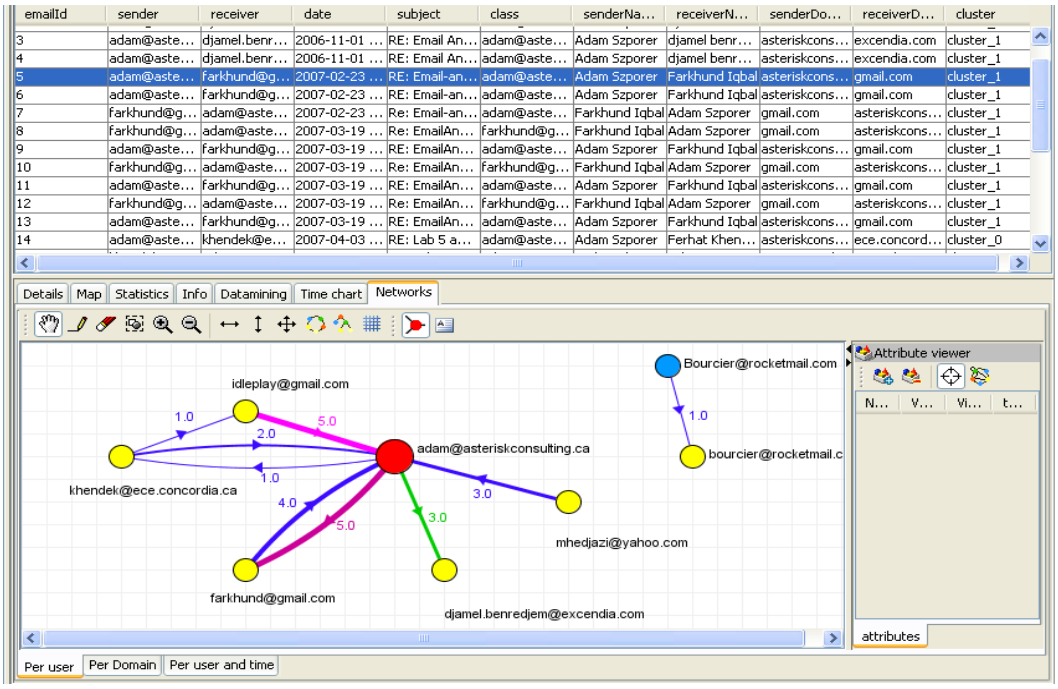


Figure 3.2: User model

Our framework provides interesting information rendering and exploration capabilities for visualizing social networks. Social networks are labeled with some simple statistics computed about the flow of messages. We use two types of graphs to depict the social network of message users. In the first graph, called *user model*, the nodes denote e-mail users and the edges denote e-mail traffic, as shown in Figure 3.2. Statistical information computed on a social network is rendered graphically using features of nodes and links: size, shape, color, thickness, etc. For instance, the thickness of the links between the nodes denotes the frequency of the messages sent and the arrow denotes the direction

of message flow from sender to recipient. Similarly, the size of a node reflects a user's frequency of messages, called degree of centrality [38, 99]. An important user, e.g., the "boss," of a group of users is represented by a bigger node. Nodes associated with users can be replaced with their photos to provide a more intuitive and elegant representation.

In the second graph, called *temporal model*, the user network is augmented with time information about e-mails, plotted to show the temporal characteristics of message flow, as shown in Figure 3.3. From this network, it is easy to identify causality effects between e-mails, for instance, the scenario in which an e-mail is received by a user, who in turn sends another e-mail at a later time. If, for example, both e-mails are classified to the same topic category, e.g., drugs, then by following the chain of the e-mails one can identify the potential collaborators.

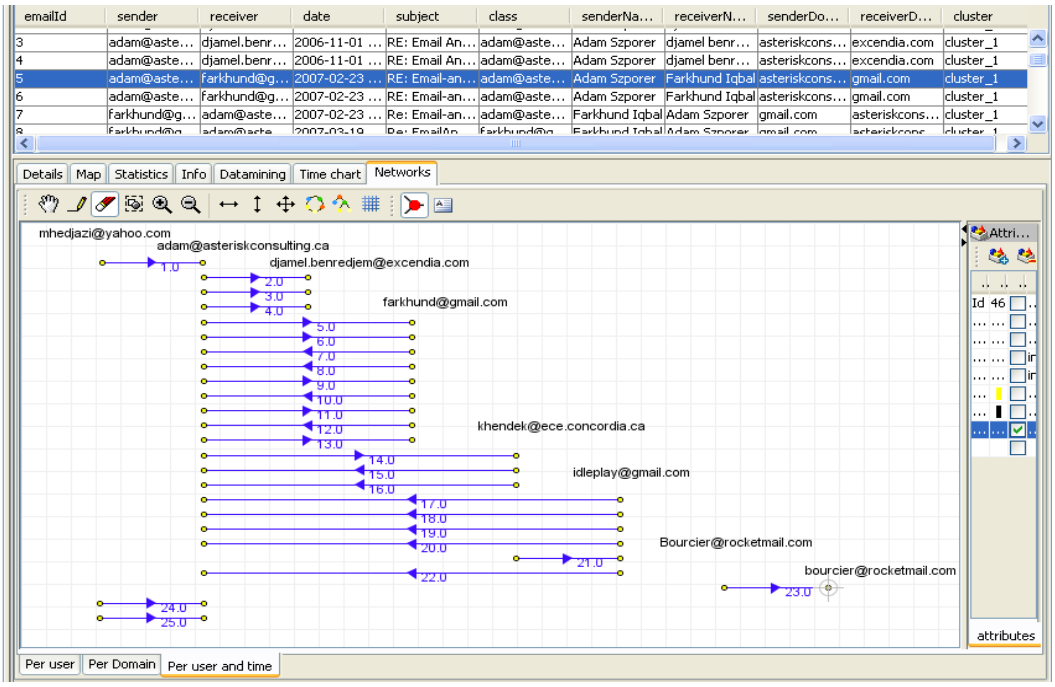


Figure 3.3: Temporal model

3.3 Geographic Localization

To understand the geographical scope of a cybercrime investigation, it is important to localize the source and destination of the given suspicious messages. This information will help an investigator in collecting additional clues about the potential suspects and their physical locations. For this, we add a geographic visualization capability in our framework, called *map viewer*, as shown in Figure 3.4. This capability can also be used to localize information related to potential suspects, e-mail servers, and e-mail flow.

The proposed *map viewer* employs the commonly used geographical localization techniques. It is a two-step process. First, the domain name of an e-mail server, extracted from an e-mail address, is translated into the corresponding IP address by using the domain name server. Second, the geographical coordinates of the e-mail server are identified by employing geographical localization techniques, provided at `http://www.geobytes.com/`. In situations where the localization fails the server is mapped to a default geographic location in the Atlantic Ocean having coordinates: latitude=0 and longitude=0. Once the physical location of each e-mail account is identified, the next step is to display them on the global map. For this, we draw an arrow from sender to recipient as shown in Figure 3.4.

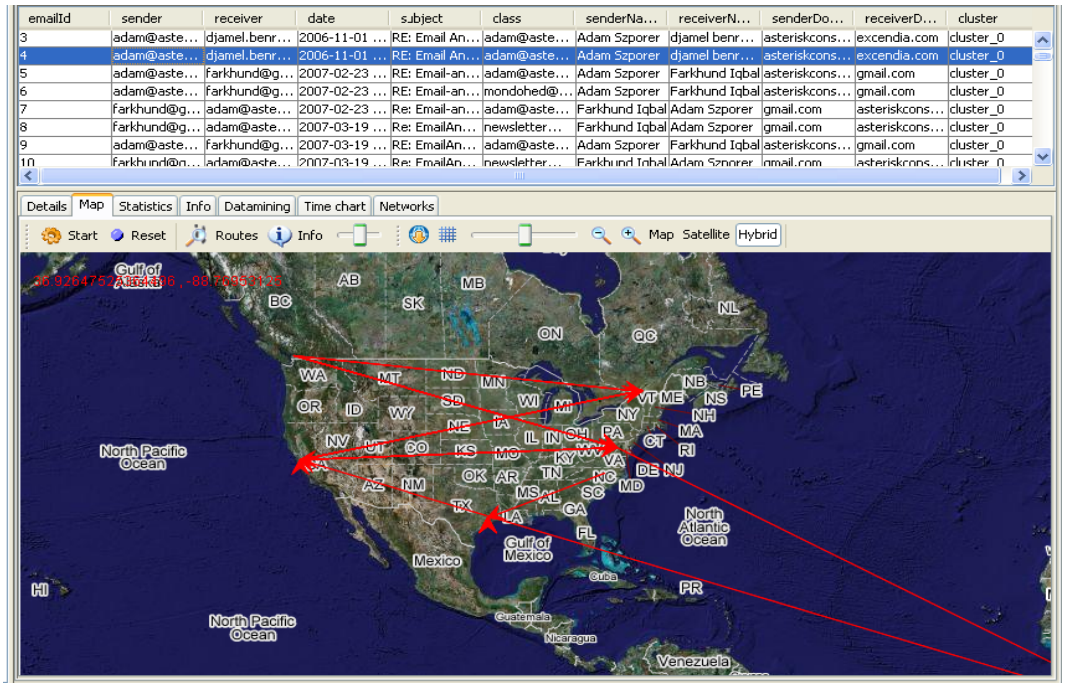


Figure 3.4: Map viewer

3.4 Text Mining

Traditional keyword searching for identifying malicious documents is inefficient and error prone due to a criminal community's use of sophisticated obfuscation techniques. Therefore, text mining, including classification and clustering, has gained great importance in the context of computer forensics. Classification or supervised learning is used to identify the class label of an unknown document based on some previously defined classes. Classification techniques have been very successful in resolving authorial disputes over poetic and historic collections. In cybercrime investigation classification techniques are used for authorship analysis of anonymous messages for identifying perpetrators involved in illegitimate activities. Clustering is an unsupervised learning process used to retrieve hidden patterns and structures [6] from a dataset without having any previous knowledge.

Unlike classification, where an unknown object is assigned to one of the predefined class, clustering is applied to identify the pertinent groups of objects based on some similarity measure. Clustering is employed for information retrieval [69] and authorship similarity detection [122]. Clustering can be applied to textual content as well as stylometric features.

In addition to header-content analysis, we also use traditional text categorization techniques [43] for message classification. In general, text classification starts by preprocessing, followed by classifier training and testing. The validated model is then employed to identify the class label of the unknown document. The class label is the topic name from a list of predefined topic categories.

Preprocessing is an essential step in most text mining processes. Preprocessing starts by message extraction, followed by cleaning, tokenization, stemming, and stopword removal. Often the available data is noisy, containing unwanted and irrelevant information. Similarly, the data need to be converted into the format acceptable by the data mining process in question. After extracting the body of an online message (e.g., an e-mail or a chat session), text written in a language other than English or French (in some cases) is discarded. E-mail attachments, chain of replied messages, and (in some cases) HTML tags are also deleted from the e-mails.

We use Java tokenizer API to convert each message μ into a set of tokens or words. The different forms of the same word appearing in a message are converted into the root word by applying stemming algorithms, e.g., Porter stemmer [81, 86]. For instance, the words write, wrote, written, and writing are converted into the word (say) write. The list

of tokens are then scanned for the all-purpose stopwords, containing function words (e.g., 'is', 'my', 'yours', and 'below'), short words (e.g., words containing 1-3 characters), punctuation, special characters, and space characters. These words usually do not contribute to the subject matter of a message and are deleted. The actual number of function words varies; [121] lists 150 while [4] mentions 303 function words.

The content-specific terms are used for features extraction. A feature is usually the relative weight calculated for each term. It can be simply the frequency of a term t_j within a message μ_i denoted by $tf_{(i,j)}$; or it can be computed by employing certain functions such as $tf-idf$, described in [61], and is given as

$$(tf-idf)_{(i,j)} = tf_{(i,j)} * idf_{(i,j)}$$

where $tf-idf_{(i,j)}$ is the weight of a term t_j within a message μ_i, $tf_{(i,j)}$ is the frequency of a term t_j within message μ_i, $idf_{(i,j)} = log(\frac{N}{df_i})$ is the inverse document frequency, N is the total number of messages, and df_i is the number of messages where the term t_i appears.

Each message μ is represented as a 'bag of words' using vector space representation [93]. Once all the messages are converted into vectors, normalization is applied to scale down the term frequencies to [0,1] to avoid overweighing one feature over another. The selected column is scanned for the maximum number and is used to divide all other members of that column.

To develop a classification model we divide the given message collection into two sets: a training set (comprising $\frac{2}{3}$ of total messages) and testing set comprising ($\frac{1}{3}$ of total messages). Each message instance of the given sample data carries a class label, representing its topic category. Common classifiers include *decision tree* [87], *neural*

networks [70], and *Support Vector Machine* [61]. The validated model is then employed for classification of a message for which the topic category is not known. Usually, the larger the training set, the better the accuracy of the model. For this purpose, we use WEKA, a data mining software toolkit [111]. Therefore, the feature vectors are converted into WEKA compatible format, Attribute-Relation File Format (ARFF), described in Section 2.2.

Sometimes an investigator is asked to analyze a given collection of anonymous documents without any prior knowledge. To initiate the process of investigation the investigator would like to identify the major topics contained in the given documents. Traditional content-based clustering can be used to first divide the messages into pertinent groups, and then tag each cluster with the most frequent words, as discussed in Section 2.4. In our framework we use three clustering algorithms: Expectation Maximization (EM), *k*-means, and bisecting *k*-means.

Once the clusters are obtained, each cluster is tagged with the high frequency words found in the respective cluster. The clusters can be used for document retrieval by matching the given keywords with the cluster labels. The matched clusters are retrieved in the order of relevance to the search criterion (query content).

3.5 Summary

In this chapter, we have presented the header-level functionalities of our framework. For instance, we have applied statistical analysis to get an overview of the given message collection. Social network analysis techniques have been used to learn about the flow of

messages between the message users. Geographical techniques have been employed to localize the users on the global map. The predictive machine learning algorithms (classification and clustering) have been applied for message classification and categorization.

Chapter 4

Writeprint Mining for Authorship Attribution

In this chapter, we develop a novel approach of frequent pattern-based writeprint creation to address two authorship problems, i.e., authorship attribution in the *usual way*, and authorship attribution by focusing on *stylistic variations*. Stylistic variation is the occasional change in the writing features of an individual with respect to the type of recipient s/he is writing to and the topic of a message. The authorship methods proposed in this chapter and in the following chapters are applicable to different types of online messages. However, for the purpose of experimentation, we use an e-mail corpus in this chapter.

The *problem of authorship attribution* in the context of online messages can be described as follows: a cyber forensic investigator wants to determine the author of a given malicious e-mail ω and has to prove that the author is likely to be one of the suspects $\{S_1, \cdots, S_n\}$. The problem is to identify the most plausible author from the suspects

$\{S_1, \cdots, S_n\}$ and to gather convincing evidence to support the finding in a court of law.

The problem of authorship identification in the context of e-mail forensics is distinct from traditional authorship problems in two ways. First, the number of potential suspects is larger and their (usually confiscated) previously written documents (e.g., e-mails), available to the investigator, are greater in number. Second, by assumption, the true author should certainly be one of the suspects.

The problem of authorship analysis becomes more challenging by taking into consideration the occasional variation in the writing style of the same person. The authorship attribution studies [34, 121] discuss contextual and temporal variation in people's writing styles, but none of these studies propose methods for capturing the stylistic variation. The writing style of a suspect may change either due to change in the context (or topic of discussion in e-mail) or the type of recipient [34]. Employing analytical techniques over the entire collection of an author's writing samples without considering the issue of *stylistic variation* (the term coined for the first time in the current study) would produce misleading results.

In this chapter, we propose a novel approach of extracting a frequent pattern-based "writeprint" to address the attribution problem in the usual way, as depicted in Figure 4.1. Then, we extend the proposed approach to address the same problem of authorship attribution with *stylistic variation* or stylistic inconsistency, as shown in Figure 4.2.

The use of *fingerprinting* techniques for identifying a potential suspect in a traditional criminal investigation process is not applicable to the digital world. However, authorship studies [20, 116] suggest that people usually leave traces of their personality

in their written work. Therefore, in cyber forensics, an investigator would like to identify the "writeprint" of an individual from his/her e-mail messages and use it for authorship attribution. The key question is:

What exactly are the patterns that can represent the writeprint of an individual?

Our insight is that the writeprint of an individual is the *combinations of features* that occur frequently in his/her written e-mail messages. The commonly used features are lexical, syntactical, structural, and content-specific attributes (see Section 2.3.1). By matching the writeprint with the malicious e-mail, the true author can be identified. Most importantly, the matched writeprint *should* provide credible evidence for supporting the conclusion. The research community [33, 104, 121] has devoted a lot of effort studying stylistic and structural features *individually*, but few have studied the *combinations* of features that form a writeprint and addressed the issue of evidence gathering.

Figure 4.1 depicts an overview of our proposed method, called *AuthorMiner1*, for addressing the usual attribution problem. We first extract the set of frequent patterns independently from the e-mail messages M_i written by suspect S_i. Though the set of frequent patterns captures the writing style of a suspect S_i, it is inappropriate to use *all* the frequent patterns to form the writeprint of a suspect S_i because another suspect, say S_j, may share some common writing patterns with S_i. Therefore, it is crucial to filter out the common frequent patterns and identify the *unique patterns* that can differentiate the writing style of a suspect from that of others. These unique patterns form the *writeprint* of a suspect.

To address the attribution problem with stylistic variation we develop an extended

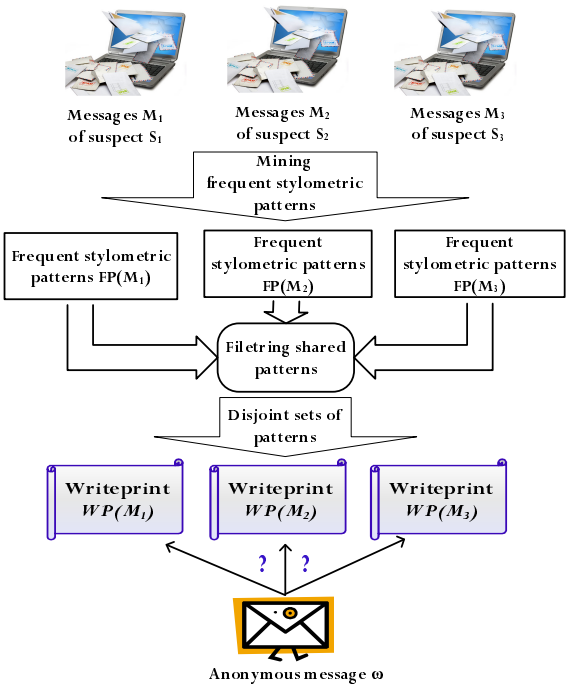


Figure 4.1: AuthorMiner1: Authorship identification without *stylistic variation*

version of the *AuthorMiner1*, called *AuthorMiner2*. An overview of AuthorMiner2 is shown in Figure 4.2 and is outlined in algorithm 4.2. First, each message collection M_i of a suspect S_i is divided into different groups $\{G_i^1, \cdots, G_i^k\}$. Second, frequent stylometric patterns $FP(G_i^g)$ from each group G_i^g are extracted. Third, the frequent patterns shared between two or more groups across all the suspects are deleted. The remaining frequent stylometric patterns form the sub-writeprint of each group G_i^g, denoted by $WP(G_i^g)$. Fourth, we identify the most plausible author S_a of ω by comparing every extracted writeprint $WP(G_i^g)$ with ω.

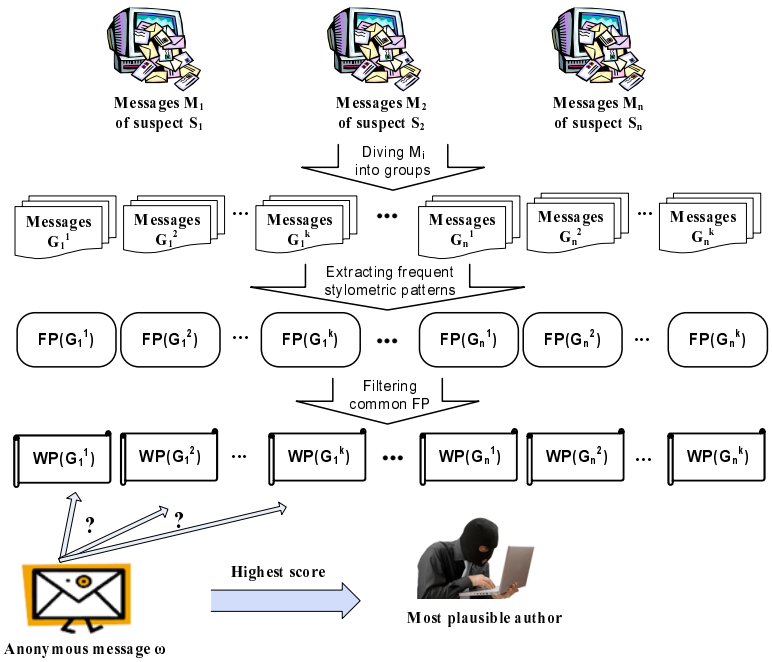


Figure 4.2: AuthorMiner2: Authorship identification with *stylistic variation*

This approach has the following merits that are not found in most of the existing works.

- *Justifiable evidence:* The write-print, represented as a set of unique patterns, is extracted from the sample documents of a particular suspect. Our method guarantees that the identified patterns are frequent in the documents of one suspect only and not frequent in others' documents. It will be difficult for an accused suspect to deny the validity of the findings. The results obtained are traceable, justifiable, and can be presented quantitatively with a statistical support. The traditional authorship identification methods, such as SVM and neural networks [104, 121], do not have the same merit.

- *Flexible writing styles:* The frequent pattern-mining technique can adopt all four

types of commonly used writing style features (described in Section 2.3.1). This flexibility is important for determining the combined effect of different features. This is much more flexible than the traditional decision tree, which primarily relies on the nodes at the top of the tree to differentiate the writing styles of all suspects.

- *Features optimization:* Unlike traditional approaches, where it is hard to determine the contribution of each feature in the authorship attribution process [34], the proposed technique is based on the distinctive patterns, the combination of features. The support associated to each pattern in the write-print set determines the contribution of each pattern.

- *Capturing inconsistent stylistics:* Our analysis shows that the writing style of a person is not usually consistent and may change depending on the recipients and the context of the message. Our proposed algorithm is able to capture the sub-stylistic attributes of an individual by employing the idea of frequent patterns. Our experimental results suggest that the identification of sub-writeprints can improve the accuracy of authorship identification. Most importantly, the sub-writeprint reveals the fine-grained writing styles of an individual that can be valuable information for an investigator.

The rest of the chapter is organized as follows: Section 4.1 formally defines the two subproblems of authorship attribution. Section 4.2 describes the building blocks of the two proposed approaches. Section 4.3 describes our proposed approaches for addressing the two subproblems. Section 4.4 evaluates the proposed two methods on a real-life e-mail dataset. Section 4.5 concludes the chapter.

4.1 Problem Statement

The problem of authorship attribution is divided into two subproblems. The first subproblem is the traditional authorship attribution problem in which we ignore the occasional change in the writing style of a person and try to extract the stylometric features from the entire message collection of a suspect. In the second subproblem, we take the style inconsistency or stylistic variation of a suspect into consideration and propose methods for dividing the messages of each suspect into different groups to capture the stylistic variation prior to apply the authorship identification process.

4.1.1 Attribution without Stylistic Variation

The *problem of authorship attribution* is to identify the true author of an anonymous message ω. The true author is assumed to be among the potential suspects $\{S_1, \cdots, S_n\}$. The investigator assumes to have access to the training samples of the suspects. In real-life investigation, the sample text messages can be obtained from the suspects' e-mail archives and chat logs on the seized personal computer, or from the e-mail service provider with warrants. The findings need to be supported with convincing arguments.

Definition 4.1.1 (Authorship attribution)**.** Let $\{S_1, \cdots, S_n\}$ be the set of suspected authors of a malicious e-mail message ω. We assume to have access to sample messages M_i, for each suspect $S_i \in \{S_1, \cdots, S_n\}$. The *problem of authorship attribution* is to identify the most plausible author S_a, from the suspects $\{S_1, \cdots, S_n\}$, whose collection of messages M_a has the "best match" with the patterns in the malicious message ω. Intuitively, a collection of messages M_i *matches* ω if M_i and ω share similar patterns of stylometric

features such as vocabulary usage. ■

The problem of authorship attribution can be refined into three subproblems: (1) To identify the writeprint $WP(M_i)$ from each set of e-mail messages $M_i \in \{M_1, \cdots, M_m\}$. (2) To determine the author of the malicious e-mail ω by matching ω with each of $\{WP(M_1), \cdots, WP(M_m)\}$. (3) To extract evidence for supporting the conclusion on authorship. The evidence has to be intuitive enough for convincing the judge and the jury in the court of law. These three subproblems summarize the challenges in typical investigation procedure. To solve subproblems (1) and (2), we first extract the set of frequent patterns $FP(M_i)$ from M_i and then filter out the patterns appearing in any other sets of e-mails M_j. For subproblem (3), the writeprint $WP(M_a)$ could serve the evidence for supporting the conclusion, where M_a is the set of e-mail messages written by the identified author S_a.

4.1.2 Attribution with Stylistic Variation

The existing authorship studies though mention about the changing style of an author either conscientiously or unconscientiously in his writing, however, they ignore it. In the current study, we define the problem of authorship attribution with focus on the problem of *stylistic variation* or *volitive stylistics*. The problem is to first isolate the different sub-styles of a suspect, capture those styles and then compute the writeprint for each sub-style called sub-writeprint of the suspect. Next, the anonymous message is compared with each sub-writeprint to identify its true author. More explicit description of the problem definition is given as follows.

Definition 4.1.2 (Authorship identification with stylistic variation)**.** Suppose $\{S_1, \cdots, S_n\}$ be a set of suspected authors of an anonymous text message ω. Let $\{M_1, \cdots, M_n\}$ be the sets of text messages previously written by suspects $\{S_1, \cdots, S_n\}$, respectively. Assume the message samples reflect the phenomenon of *stylistic variation*, which means the collected messages contain different topics and are written to different types of recipients, e.g., co-workers and friends. Further, assuming the number of messages of each set M_i, denoted by $|M_i|$, is reasonably large (say $>$30). The problem is to first divide messages M_i of each suspect $S_i \in \{S_1, \cdots, S_n\}$ into k different groups $\{G_i^1, \cdots, G_i^k\}$ and then apply the attribution problem to identify the most plausible suspect $S_a \in \{S_1, \cdots, S_n\}$. The suspected author S_a is the one whose (at least one) sub-writeprint has the "best match" with the features in ω.

4.2 Building Blocks of the Proposed Approach

The core concepts or the building blocks of our proposed approach are: features extraction and feature discretization. The extracted features are used to identify frequent stylometric patterns and convert them into the writeprint. A detailed description of these concepts is given below.

4.2.1 Feature Extraction

The feature extraction starts by message extraction followed by cleaning, tokenization, and stemming, as discussed in Section 3.4. There are more than a thousand stylometric features used so far in different studies [4, 121]. As listed in Table 4.1 and Table 4.2, we carefully select 285 features in our study. In general, there are three types of features. The first type is a numerical value, e.g., the frequencies of some individual characters, punctuations, and special characters. To avoid the situation where very large values overweigh other features, we apply normalization to scale down all the numerical values to $[0, 1]$.

The second type is a boolean value, e.g., to check whether or not an e-mail contains a reply message. The third type of features is computed by taking as input some other lexical functions such as vocabulary richness, indexed at 93-98 in Table 4.1. Most of these features are computed in terms of vocabulary size $V(N)$ and text length N [106]. When feature extraction is done, each e-mail is represented as a vector of feature values. In this thesis we focus on using structural features as they play a significant role in distinguishing writing styles.

Short words comprising of 1-3 characters (such as ‘is’, ‘are’, ‘or’, ‘and’, etc.) are mostly context-independent and are counted together. Frequencies of words of various lengths 1-30 characters, indexed at 58-87 in Table 4.1, are counted separately. Hepax Legomena and Hapax dislegomena are the terms used for once-occurring and twice-occurring words. As mentioned earlier, we have used more than 150 function words, listed in Appendix I.

We also check whether an e-mail has welcoming and/or farewell greetings. Paragraph separator can be a blank line or just a tab/indentation or there may be no separator between paragraphs.

Table 4.1: Lexical and syntactic features

Feature Type	**Feature Name**
Lexical	1. Character count including space characters (M) 2. Ratio of digits to M 3. Ratio of letters to M 4. Ratio of uppercase letters to M 5. Ratio of spaces to M 6. Ratio of tabs to M 7-32. Alphabet frequency (A-Z) (26 features) 33-53. Occurrences of special characters: < > % | { } [ ] / \ @ # ~ + - * $ ^ & _ $ \div$ (21 features) 54. Word count (W) 55. Average word length 56. Average sentence-length in terms of characters 57. Ratio of short words (1-3 characters) to W 58-87. Ratio of word length frequency distribution to W (30 features) 88. Ratio of function words to W 89. Vocabulary richness, i.e., T/W 90. Ratio of Hapax legomena to W 91. Ratio of Hapax legomena to T 92. Ratio of Hapax dislegomena to W 93. Guirad's R 94. Herdan's C 95. Herdan's V 96. Rubet's K 97. Maas' A 98. Dugast's U
Syntactic	99-106. Occurrences of punctuations , . ? ! : ; ' " (8 features) 107. Ratio of punctuations with M 108-257. Occurrences of function words (150 features)

Thirteen content-specific terms (273-285) are selected from the Enron e-mail corpus [1] by applying content-based clustering. Each message is represented as a feature

[1] http://www-2.cs.cmu.edu/~enron/

vector using vector space model, as shown in Table 4.3. This table represents 10 sample messages, where each row represents one e-mail message.

Table 4.2: Structural and domain-specific features

Feature Type	**Feature Name**
Structural	258. Ratio of blank lines/total number of lines within e-mail 259. Sentence count 260. Paragraph count 261. Presence/absence of greetings 262. Has tab as separators between paragraphs 263. Has blank line between paragraphs 264. Presence/absence of separator between paragraphs 265. Average paragraph length in terms of characters 266. Average paragraph length in terms of words 267. Average paragraph length in terms of sentences 268. Contains Replied message 269. Position of replied message in the e-mail 270. Use e-mail as a signature 271. Use telephone as signature 272. Use URL as a signature
Domain-specific	273-285. deal, HP, sale, payment, check, windows, software, offer, Microsoft, meeting, conference, room, report (13 features)

One may first apply *feature selection* [75] as a preprocessing step to determine a subset of stylometric features that can discriminate the authors. There are two general approaches [98]: *Forward selection* starts with no features and, at each step, adds the feature that decreases the error the most until any further addition does not decrease the error significantly. *Backward selection* starts with all the features and, at each step, remove the one that decreases the error the most until any further removal increases the error significantly. These approaches consider only one attribute at a time. In contrast, our proposed approach employs the notion of frequent stylometric patterns that capture the combined effect of features. Irrelevant features will not be frequent in our approach. Thus, there is

no need to apply feature selection. More importantly, feature selection does not guarantee the property of uniqueness among the writeprints of the suspects.

Table 4.3: Stylometric feature vectors (prior to discretization)

Messages (μ)	**Feature X**	**Feature Y**	**Feature Z**
μ_1	0.130	0.580	0.555
μ_2	0.132	0.010	0.001
μ_3	0.133	0.0124	0.123
μ_4	0.119	0.250	0.345
μ_5	0	0.236	0.532
μ_6	0.150	0.570	0.679
μ_7	0	0.022	0.673
μ_8	0.865	0.883	0.990
μ_9	0.137	0.444	0.494
μ_{10}	0.0	0.455	1.000

4.2.2 Feature Discretization

The feature vectors, extracted in the previous step, contain numeric values. To extract frequent patterns from the message dataset, we apply Apriori algorithm [7]. For this, we need to transform the numeric feature values into boolean type indicating the presence or absence of a feature within a message. We discretize each feature $F_a \in \{F_1, \cdots, F_g\}$ into a set of intervals $\{\imath_1, \cdots, \imath_h\}$, called *feature items*. Common discretization techniques are:

- *Equal-width discretization*, where the size of each interval is the same.
- *Equal-frequency discretization*, where each interval has approximately the same number of records assigned to it.

- *Clustering-based discretization*, where clustering is performed on the distance of neighboring points.

Due to the small size of an e-mail message, most feature values fall into the beginning of an interval and need to be discretized in a more dynamic way. Our initial experimental results indicate that the value of most features are close to zero with very few features having larger values. Therefore, employing *equal-width discretization*, and/or *equal-frequency discretization* is not a good choice while the *clustering-based discretization* method is complex and computationally expensive. To fit the niche, we have developed a new discretization mechanism called *controlled binary split* which has substantially improved the results as compared to our initial study [60].

In the proposed technique, we successively split the feature value into two intervals and check if the number of feature occurrences is less than the user specified threshold or not. The binary splitting continues until all the feature values are discretized. The normalized feature frequency, found in a message, is then matched with these intervals. A boolean '1' is assigned to the feature item if the interval contains the normalized feature frequency; otherwise a '0' is assigned.

Example 4.2.1. Consider Table 4.4, which contains 10 e-mail messages. Let us assume that $\{X, Y, Z\}$ represent the set of features extracted from these messages. Next, each feature is converted into feature items by applying discretization. For example, feature X having normalized values in the range $[0, 1]$ and suppose the user threshold is 5%, i.e., the splitting continues until each interval contains at most 5% of the total number of feature occurrences. Feature X is discretized into three intervals $X_1 = [0, 0.120]$,

Table 4.4: Stylometric feature vectors (after discretization)

Messages (μ)	**Feature X**			**Feature Y**		**Feature Z**	
	X_1	X_2	X_3	Y_1	Y_2	Z_1	Z_2
μ_1	0	1	0	0	1	0	1
μ_2	0	1	0	1	0	1	0
μ_3	0	1	0	1	0	1	0
μ_4	1	0	0	1	0	1	0
μ_5	0	0	0	1	0	0	1
μ_6	0	0	1	0	1	0	1
μ_7	0	0	0	1	0	0	1
μ_8	0	0	1	0	1	0	1
μ_9	0	1	0	1	0	1	0
μ_{10}	0	0	0	1	0	0	1

$X_2 = (0.120, 0.140]$, and $X_3 = (0.140, 1.000]$, representing three feature items. Similarly, features Y and Z are discretized into $Y_1 = [0, 0.500]$, $Y_2 = (0.500, 1]$, $Z_1 = [0, 0.500]$, and $Z_2 = (0.500, 1]$, respectively. The message μ_1 containing features $X = 0.130$, $Y = 0.250$, and $Z = 0.020$ can be represented as a feature vector $\langle X_2, Y_2, Z_2 \rangle$. ■

4.2.3 Frequent Stylometric Patterns

Intuitively, the stylometric patterns or the writing style patterns in an ensemble of e-mail messages M_i (written by suspect S_i) is a combination of feature items that *frequently* occurs in M_i. We concisely model and capture such frequent patterns by the concept of *frequent itemset* [7] described as follows.

Let $U = \{\mathfrak{u}_1, \cdots, \mathfrak{u}_u\}$ denote the universe of all *stylometric feature items*. Let M_i be a set of e-mail messages where each message $\mu \in M_i$ is represented as a set of stylometric feature items such that $\mu \subseteq U$. A text message μ contains a stylometric feature item $\mathfrak{u}_j$ if the numerical feature value of the message μ falls within the interval of $\mathfrak{u}_j$. The writing

style features of some sample messages are represented as vectors of feature items in Table 4.5.

Let $P \subseteq U$ be a set of stylometric feature items called a *stylometric pattern*. A text message μ *contains* a stylometric pattern P if $P \subseteq \mu$. A stylometric pattern that contains κ stylometric feature items is a κ-*pattern*. For example, the stylometric pattern $P = \{\iota_1, \iota_4, \iota_6\}$ is a 3-pattern. The *support* of a stylometric pattern P is the percentage of text messages in M_i that contains P. A stylometric pattern P is *frequent* in a set of messages M_i if the support of P is greater than or equal to a user-specified minimum support threshold.

Definition 4.2.1 (Frequent stylometric pattern)**.** Let M_i be the set of text messages written by suspect S_i. Let $support(P|M_i)$ be the percentage of text messages in M_i that contain the pattern P, where $P \subseteq U$. A pattern P is a *frequent stylometric pattern* in M_i if $support(P|M_i) \geq min\_sup$, where the minimum support threshold min_sup is a real number in an interval of $[0,1]$. ■

The writing style of a suspect S_i is represented as a set of frequent stylometric patterns, denoted by $FP(M_i) = \{P_1, \cdots, P_l\}$, extracted from his/her set of text messages M_i.

Example 4.2.2. Consider the messages, represented as vectors of feature items, in Table 4.5. Suppose the user-specified threshold $min\_sup = 0.3$, which means that a stylometric pattern $P = \{\iota_1, \cdots, \iota_e\}$ is frequent if at least 3 out of the 10 e-mails contain all feature items in P. For instance, $\{X_1\}$ is not a frequent stylometric pattern because it has support 2/10=0.2. The feature item $\{X_2\}$ is a frequent stylometric 1-pattern because it has

support 0.4. Similarly, $\{X_2, Y_1\}$ is a frequent stylometric 2-pattern because it has support 0.4. Likewise, $\{X_2, Y_1, Z_1\}$ is a frequent stylometric 3-pattern because it has support 0.3. Example 8.2.1 shows how to efficiently compute all frequent patterns. ■

Table 4.5: Message representation in terms of feature items

Messages (μ)	Feature items
μ_1	$\{X_2, Y_2, Z_2\}$
μ_2	$\{X_2, Y_1, Z_1\}$
μ_3	$\{X_2, Y_1, Z_1\}$
μ_4	$\{X_1, Y_1, Z_1\}$
μ_5	$\{Y_1, Z_2\}$
μ_6	$\{X_3, Y_2, Z_2\}$
μ_7	$\{Y_1, Z_2\}$
μ_8	$\{X_3, Y_2, Z_2\}$
μ_9	$\{X_2, Y_1, Z_1\}$
μ_{10}	$\{X_1, Y_1, Z_2\}$

4.2.4 Writeprint

In forensic science, an individual can be uniquely identified by his/her fingerprint. In cyber forensics, can we identify the "writeprint" of an individual from his/her e-mails? We do not claim that the identified writeprint in this study can uniquely distinguish every individual in the world, but the identified writeprint is accurate enough to uniquely identify the writing pattern of an individual among the suspects $\{S_1, \cdots, S_n\}$ because common patterns among the suspects are filtered out and will not become part of the writeprint.

The notion of frequent pattern in Definition 8.1.1 captures the writing patterns of a suspect. However, two suspects S_i and S_j may share some similar writing patterns. Therefore, it is important to filter out the common frequent patterns and retain the frequent

patterns that are unique to each suspect. This leads us to the notion of writeprint.

Intuitively, a writeprint can uniquely represent the writing style of a suspect S_i if its patterns are found *only* in the e-mails written by S_i, but not in any other suspect's e-mails. In other words, the writeprint of a suspect S_i is a collection of frequent patterns that are frequent in the e-mail messages M_i written by S_i but not frequent in the messages M_j written by any other suspect S_j where $i \neq j$.

Definition 4.2.2 (writeprint)**.** A *writeprint*, denoted by $WP(M_i)$, is a set of patterns where each pattern P has $support(P|M_i) \geq min\_sup$ and $support(P|M_j) < min\_sup$ for any M_j where $i \neq j$, min_sup is a user-specified minimum threshold. In other words, $WP(M_i) \subseteq FP(M_i)$, and $WP(M_i) \cap WP(M_j) = \emptyset$ for any $1 \leq i, j \leq n$ and $i \neq j$. ■

4.3 Proposed Approaches

The proposed solution is divided into two parts. The first part, called AuthorMiner1, addresses the traditional attribution problem without the consideration of stylistic variation while the second part, called AuthorMiner2, addresses the attribution problem with the consideration of stylistic variation. Detailed description of the two components is given in the following two subsections.

4.3.1 AuthorMiner1: Attribution without Stylistic Variation

Algorithm 4.1 presents a novel data mining method, called *AuthorMiner1*, for determining the authorship of a malicious e-mail message ω from a group of suspects $\{S_1, \cdots, S_n\}$

based on the extracted features of their previously written e-mail messages $\{M_1, \cdots, M_n\}$. In this section, an e-mail message is represented by a set of feature items. We summarize the algorithm in the following three phases followed by a detailed description of each phase.

Require: An anonymous message ω.
Require: Sets of messages $\{M_1, \cdots, M_n\}$, written by $\{S_1, \cdots, S_n\}$.
/* Mining frequent stylometric patterns */
for each $M_i \in \{M_1, \cdots, M_n\}$ **do**
  extract frequent stylometric patterns $FP(M_i)$ from M_i;
end for
/* Filtering out common frequent patterns */
for each $FP(M_i) \in \{FP(M_1), \cdots, FP(M_n)\}$ **do**
  for each $FP(M_j) \in \{FP(M_{i+1}), \cdots, FP(M_n)\}$ **do**
    for each frequent pattern $P_x \in FP(M_i)$ **do**
      for each frequent pattern $P_y \in FP(M_j)$ **do**
        if $P_x = P_y$ **then**
          $FP(M_i) \leftarrow FP(M_i) - P_x$;
          $FP(M_j) \leftarrow FP(M_j) - P_y$;
        end if
      end for
    end for
  end for
  $WP(M_i) \leftarrow Disjoint\ frequent\ patterns(M_i)$;
end for
/* Identifying author */
$highest\_score \leftarrow -1$;
for all $WP(M_i) \in \{WP(M_1), \cdots, WP(M_n)\}$ **do**
  if $Score(\omega \approx WP(M_i)) > highest\_score$ **then**
    $highest\_score \leftarrow Score(\omega \approx WP(M_i))$;
    $author \leftarrow S_i$;
  end if
end for
return $author$;

Algorithm 1: AuthorMiner1

Phase 1: Mining frequent patterns (Lines 1-3). Extract the frequent patterns $FP(M_i)$ from each collection of e-mail messages M_i written by suspect S_i. The extracted frequent patterns capture the writing style of a suspect.

Phase 2: Filtering common frequent patterns (Lines 4-16). Though $FP(M_i)$ may capture the writing patterns of suspect S_i, $FP(M_i)$ may contain frequent patterns that are shared by other suspects. Therefore, Phase 2 removes the common frequent patterns. Specifically, a pattern P in $FP(M_i)$ is removed if *any* other $FP(M_j)$ also contains P, where $i \neq j$. The remaining frequent patterns in $FP(M_i)$ form the writeprint $WP(M_i)$ of suspect S_i. When this phase completes, we have a set of writeprints $\{WP(M_1), \cdots, WP(M_n)\}$ of suspects $\{S_1, \cdots, S_n\}$. Figure 4.1 illustrates that the writeprint $WP(M_2)$ comes from $FP(M_2)$ by filtering out the frequent patterns shared by $FP(M_1)$, $FP(M_2)$, and/or $FP(M_3)$.

Phase 3: Identifying author (Lines 17-24). Compare the malicious e-mail message ω with each writeprint $WP(M_i) \in \{WP(M_1), \cdots, WP(M_n)\}$ and identify the most similar writeprint that matches ω. Intuitively, a writeprint $WP(M_i)$ is similar to the e-mail message ω if many frequent patterns in $WP(M_i)$ can be found in ω. Our insight is that the frequent patterns are not equally important. Their importance is reflected by their $supprt(P|M_i)$; therefore, we derive a score function $Score(\omega \approx WP(M_i))$ to measure the weighted similarity between the e-mail message ω and the frequent patterns in $WP(M_i)$. The suspect S_a of writeprint $WP(M_a)$, which has the highest $Score(\omega \approx WP(M_i))$, is classified to be the author of the malicious e-mail message ω.

Mining Frequent Stylometric Patterns (Lines 1-3): Lines 1-3 mine the frequent patterns $FP(M_i)$ from each collection of e-mail message $M_i \in \{M_1, \cdots, M_n\}$, for $1 \leq i \leq n$. There are many data mining algorithms for extracting frequent patterns, for example, Apriori [7], FP-growth [51], and ECLAT [118]. Below, we provide an overview of the Apriori algorithm which has been previously applied to various text mining tasks [46,57].

Apriori is a level-wise iterative search algorithm that uses frequent κ-patterns to explore the frequent $(\kappa+1)$-patterns. First, the set of frequent 1-patterns is found by scanning the e-mail messages M_i, accumulating the support count of each feature item, and collecting the feature item ι that has $support(\iota|M_i) \geq min\_sup$. The resulting frequent 1-patterns are then used to find frequent 2-patterns, which are then used to find frequent 3-patterns, and so on, until no more frequent κ-patterns can be found. The generation of frequent $(\kappa+1)$-patterns from frequent κ-patterns is based on the following Apriori property.

Property 4.3.1 (Apriori property)**.** All nonempty subsets of a frequent pattern must also be frequent. ■

By definition, a pattern P is not frequent if $support\ (P|M_i) < min\_sup$. The above property implies that adding a feature item ι to a non-frequent pattern P will never make it more frequent. Thus, if a κ-pattern P is not frequent, then there is no need to generate $(\kappa+1)$-pattern $P \cup \iota$ because $P \cup \iota$ is also not frequent. The following example shows how the Apriori algorithm exploits this property to efficiently extract all frequent patterns. Refer to [7] for a formal description.

Example 4.3.1. Consider Table 4.5 with $min\_sup = 0.3$. First, identify all frequent 1-patterns by scanning the database once to obtain the support of every feature item. The feature items having support ≥ 0.3 are frequent 1-patterns, denoted by L_1 = $\{\{X_2\}, \{Y_1\}, \{Z_1\}, \{Z_2\}\}$. Then, join L_1 with itself, i.e., $L_1 \bowtie L_1$, to generate the candidate list ℓ_2 = $\{\{X_2,Y_1\}, \{X_2,Z_1\}, \{X_2,Z_2\}, \{Y_1,Z_1\}, \{Y_1,Z_2\}, \{Z_1,Z_2\}\}$ and scan the database once to obtain the support of every pattern in ℓ_2. Identify the frequent 2-patterns, denoted by L_2 = $\{\{X_2,Y_1\}, \{X_2,Z_1\}, \{Y_1,Z_1\}, \{Y_1,Z_2\}\}$. Similarly, perform $L_2 \bowtie L_2$ to generate ℓ_3 and scan the database once to identify the frequent 3-patterns which is $L_3 = \{X_2,Y_1,Z_1\}$. The finding of each set of frequent κ-patterns requires one full scan of the feature items in Table 4.5. ■

Filtering Common Patterns (Lines 4-16): This phase filters out the common frequent patterns among $\{FP(M_1),\cdots,FP(M_n)\}$. The general idea is to compare every frequent pattern P_x in $FP(M_i)$ with every frequent pattern P_y in *all* other $FP(M_j)$, and to remove them from $FP(M_i)$ and $FP(M_j)$ if P_x and P_y are the same. The computational complexity of this step is $O(|FP(M)|^n)$, where $|FP(M)|$ is the number of frequent patterns in $FP(M)$ and n is the number of suspects. The remaining frequent patterns in $FP(M_i)$ form the writeprint $WP(M_i)$ of suspect S_i.

Example 4.3.2. Suppose there are three suspects S_1, S_2, and S_3 having three sets of e-mail messages M_1, M_2, and M_3 respectively, as depicted in Figure 4.1. Let $FP(M_1) = \{\{X_1\}, \{Y_1\}, \{Z_2\}, \{X_1,Y_1\}, \{X_1,Z_2\}, \{Y_1,Z_2\}, \{X_1,Y_1,Z_2\}\}$ be the frequent patterns of S_1. Let $FP(M_2)$ = $\{\{X_2\}, \{Y_1\}, \{Z_1\}, \{Z_2\}, \{X_2,Y_1\}, \{X_2,Z_1\}, \{Y_1, Z_1\}, \{Y_1, Z_2\}, \{X_2,Y_1,Z_1\}\}$ be the set of frequent patterns of S_2, as given in Example 8.2.1. Let $FP(M_3) = \{\{X_1\},$

$\{Y_3\}$, $\{Z_2\}$, $\{X_1,Y_3\}$, $\{X_1,Z_2\}$, $\{Y_3,Z_2\}$, $\{X_1,Y_3,Z_2\}\}$ be the set of frequent patterns of S_3. Then, we discard $\{X_1\}$, $\{Y_1\}$, $\{Z_2\}$, $\{X_1,Z_2\}$, $\{Y_1,Z_2\}$ as they are shared by two or more suspects. The remaining frequent patterns form the writeprints of the suspects: $WP(M_1) = \{\{X_1,Y_1\}, \{X_1,Y_1,Z_2\}\}$, $WP(M_2) = \{\{X_2\}, \{Z_1\}, \{X_2,Y_1\}, \{X_2,Z_1\}, \{Y_1,Z_1\}, \{X_2,Y_1,Z_1\}\}$, and $WP(M_3) = \{\{Y_3\}, \{X_1,Y_3\}, \{Y_3,Z_2\}, \{X_1,Y_3,Z_2\}\}$. ■

Identifying Author (Lines 17-24): Lines 17-24 determine the author of malicious e-mail message ω by comparing ω with each writeprint $WP(M_i) \in \{WP(M_1), \cdots, WP(M_n)\}$ and identifying the most similar writeprint to ω. Intuitively, a writeprint $WP(M_i)$ is similar to ω if many frequent patterns in $WP(M_i)$ matches the style in ω. Formally, a frequent pattern P *matches* ω if ω contains every feature item in P.

Equation 4.1 shows the score function that quantifies the similarity between the malicious message ω and a writeprint $WP(M_i)$. The frequent patterns are not equally important, and their importance is reflected by their support in M_i, i.e., the percentage of e-mail messages in M_i sharing such combination of features. Thus, the score function accumulates the support of a frequent pattern and divides the result by the number of frequent patterns in $WP(M_i)$ to normalize the factor of different sized $WP(M_i)$.

$$Score(\omega \approx WP(M_i)) = \frac{\sum_{j=1}^{\rho} support(MP_j|M_i)}{|WP(M_i)|} \tag{4.1}$$

where $MP = \{MP_1, \cdots, MP_\rho\}$ is a set of matched patterns between $WP(M_i)$ and the malicious e-mail message ω. The score is a real number within the range $[0,1]$. The higher the score means the higher the similarity between the writeprint and the malicious e-mail

message ω. The suspect having the writeprint with the highest score is the author of the malicious e-mail ω.

Example 4.3.3. Let the patterns found in the malicious e-mail message ω be $\{X_2, Y_1, Z_1\}$ and $\{X_1, Y_1, Z_2\}$. Comparing them to the writeprints in Example 4.3.4, we notice that the first pattern matches to a pattern in $WP(M_2)$ while the second pattern matches to a pattern in $WP(M_1)$. The score calculated according to Equation 4.1 is higher for $WP(M_1)$ because $|WP(M_1)| < |WP(M_2)|$. As a result, the message ω is most similar to $WP(M_1)$, suggesting that S_1 is its author. ■

In the unlikely case that multiple suspects have the same highest score, AuthorMiner1 returns the suspect whose the number of matched patterns $|MP|$ is the largest. In case multiple suspects have the same highest score and the same number of matched patterns, AuthorMiner1 returns the suspect whose the size of matched k-pattern is the largest because having a match on large sized frequent stylometric k-pattern implies a strong match. To facilitate the evaluation procedure in our experiment, the method presented here is designed to return only one suspect. In the actual deployment of the method, a more preferable solution is to return a list of suspects ranked by their scores, followed by the number of matched patterns and the size of the largest matched pattern.

4.3.2 AuthorMiner2: Attribution with Stylistic Variation

In the *AuthorMiner2*, we focus on the occasional change in the writing style of individuals due to the change in the context and/or target recipient. The change may occur both in the contents as well as in the style markers. For instance, e-mails that a person writes

to his job colleagues are more formal than what he writes to his family members and friends. Co-workers of a financial company may write more about meetings, promotion schemes, customer problems and solutions, salaries, and bonuses. E-mails exchanged among friends may contain discussion about trips, visits, funny stories, and jokes.

The writing style features like the selection and distribution of function words and punctuation may be different in different contexts. Moreover, a person may be more formal and careful in using structural features like the greeting and farewell comments in e-mails written to his "boss". One may prefer to put complete signatures including his designation and contact information in his job communication. More importantly, malicious e-mails are mostly anonymous and will devoid of such traceable information.

In fact, information (topic words and stylometric features) extracted from malicious messages is overshadowed by regular messages as the malicious messages are usually much fewer in number than the regular messages. The analytical techniques employed over such intermingled writing samples would produce misleading results.

To address the authorship problem in Definition 4.1.2, we propose the algorithm, called *AuthorMiner2*, to identify the author of an anonymous message ω from the suspects $\{S_1, \cdots, S_n\}$, based on the writeprints extracted from their previously written messages $\{M_1, \cdots, M_n\}$. AuthorMiner2 is employed to capture the different writing styles, called sub-styles, of a person, the authorship identification accuracy can be improved. Our experimental results support the hypothesis and suggest that the author identification accuracy of AuthorMiner2 is higher than AuthorMiner1. Most importantly, AuthorMiner2 can be employed to concisely present the fine-grained writing styles of an individual.

Figure 4.2 shows an overview of AuthorMiner2 in four steps. Step 1 groups the e-mail messages M_i (of suspect S_i) by the types of message recipients. The recipient type is identified by using different parameters, e.g., e-mail address domains. Each set of training sample messages M_i is divided into groups $\{G_i^1, \cdots, G_i^k\}$. Step 2 extracts the frequent stylometric patterns $FP(G_i^g)$ from each group $G_i^g \in \{G_i^1, \cdots, G_i^k\}$. Step 3 filters out the common frequent stylometric patterns shared between any two of the groups across all suspects. The remaining frequent stylometric patterns form the writeprint of each group G_i^g, denoted by $WP(G_i^g)$. Step 4 identifies the most plausible author S_a of ω by comparing each extracted writeprint $WP(G_i^g)$ with ω. Detailed description of each step in Algorithm 4.2 is given below.

Grouping Messages: Step 1 (Lines 1-2 in Algorithm 4.2) divides messages M_i of each suspect S_i into different groups $\{G_i^1, \cdots, G_i^k\}$. Grouping is done on the basis of e-mail body as well as e-mail header information. Headers usually contain sender/recipient address, time stamp, and path traveled by a message. To perform the first type of grouping, we employ clustering techniques.

Grouping based on Message Body: We apply two types of clustering: content-based and stylometry-based. In content-based clustering, the messages are divided into different groups based on the topic of discussion [69]. Stylometry-based clustering, on the other hand, is used to divide messages into different groups; each group containing similar patterns of writing style features [13].

Input: An anonymous message ω
Input: Messages $\{M_1, \cdots, M_n\}$ by $\{S_1, \cdots, S_n\}$.

for all $M_i \in \{M_1, \cdots, M_n\}$ **do**
  Divide M_i into groups $\{G_i^1, \cdots, G_i^k\}$;
  for all $G_i^g \in \{G_i^1, \cdots, G_i^k\}$ **do**
    extract frequent stylometric patterns $FP(G_i^g)$ from G_i^g;
  end for
end for
for all $M_i \in \{M_1, \cdots, M_n\}$ **do**
  for all $G_i^g \in \{G_i^1, \cdots, G_i^k\}$ **do**
    for all $M_j \in \{M_{i+1}, \cdots, M_n\}$ **do**
      for all $G_j^h \in \{G_j^1, \cdots, G_j^k\}$ **do**
        if $G_i^g \neq G_j^h$ **then**
          for all frequent stylometric pattern $P_x \in FP(G_i^g)$ **do**
            for all frequent stylometric pattern $P_y \in FP(G_j^h)$ **do**
              if $P_x = P_y$ **then**
                $FP(G_i^g) \leftarrow FP(G_i^g) - P_x$;
                $FP(G_j^h) \leftarrow FP(G_j^h) - P_y$;
              end if
            end for
          end for
        end if
      end for
    end for
    $WP(G_i^g) \leftarrow Disjoint\ set\ of\ FP(G_i^g)$;
  end for
end for
$highest\_score \leftarrow -1$;
for all $M_i \in \{M_1, \cdots, M_n\}$ **do**
  for all $G_i^g \in \{G_i^1, \cdots, G_i^k\}$ **do**
    if $Score(\omega \approx WP(G_i^g) > highest\_score$ **then**
      $highest\_score \leftarrow Score(\omega \approx WP(G_i^g))$;
      $author \leftarrow S_i$;
    end if
  end for
end for
return *author*;

Algorithm 2: *AuthorMiner2*

The process of clustering in both cases is the same. The difference is in the details of the preprocessing and feature extraction phase. In content-based clustering the preprocessing step is similar to the usual text mining process where the style markers (function words and punctuations), white and blank spaces are deleted along with other irrelevant parts of a document. The remaining content is tokenized and stemmed to obtain a list of topic words. The preprocessing phase in stylometry-based clustering is complex where most of the message content, including topic words and style markers, are used as features. Once all the e-mail messages of each author are converted into feature vectors, clustering is applied. We use three clustering algorithms: Expectation Maximization (EM), k-means, and bisecting k-means. Clustering is applied to e-mails of each author independently. The resultant clusters of each suspect S_i, are labeled as $\{G_i^1, \cdots, G_i^k\}$. Similarly, e-mail messages of S_j are clustered separately into clusters $\{G_j^1, \cdots, G_j^k\}$.

Grouping based on Message Header: We divide e-mail messages of each suspect into different groups based on header-content including *e-mail recipient* and *e-mail time stamp*. The intuition behind using the time stamp for grouping is that some researchers, like J. Stolfo et al. [101], believe that people behave differently at different times of the day and night.

People usually communicate with different categories of people at different times. For instance, most of the e-mails that a person writes during day time are exchanged with his/her co-workers. Similarly, e-mail messages written in the evening may be exchanged with his/her family members and friends. Likewise, very few of the e-mail messages that are exchanged at midnight may be written to one's job colleagues. For simplicity, we

divide the 24 hours into three time brackets: morning, evening, and night. Therefore, e-mails of a sender are divided into three categories: e-mails sent in the morning, e-mails sent in the evening, and those sent at night.

Extracting Frequent Stylometric Patterns: Step 2 (Lines 3-6 in Algorithm 4.2) extracts the frequent stylometric patterns from each group G_i^g for each message set M_i of suspect S_i. Frequent stylometric patterns $\{FP(G_i^1, \cdots, FP(G_i^k)\}$ from message subsets $\{G_i^1, \cdots, G_i^k\}$ of suspect S_i are extracted by using the technique described in Section 4.3.1.

Filtering Common Stylometric Patterns: Step 3 (Lines 7-25 in Algorithm 4.2) filters out the common stylometric frequent patterns between any two sets $FP(G_i^g)$ and $FP(G_j^h)$ where $i \neq j$. As described in Section 4.3.1, the general idea is to compare every frequent pattern P_x in $FP(G_i^g)$ with each frequent pattern P_y in all other sets, e.g., $FP(G_j^h)$, and to remove them from $FP(G_i^g)$ and $FP(G_j^h)$ if P_x and P_y are the same. The computational complexity of this step is $O(|\cup FP(G_i^g)|^2)$, where $|\cup FP(G_i^g)|$ is the total number of stylometric frequent patterns. The remaining stylometic frequent patterns in $FP(G_i^g)$ represents a sub-writeprint $WP(G_i^g)$ of suspect S_i. A suspect S_i may have multiple sub-writeprints, denoted by $\{WP(G_i^1), \cdots, WP(G_i^k)\}$ depending on how the messages are grouped in Step 1.

Example 4.3.4. Suppose there are two suspects S_1 and S_2 having two sets of text messages M_1 and M_2, respectively, where M_1 is divided into groups G_1^1 and G_1^2, and M_2 is divided into groups G_2^1 and G_2^2. Suppose $FP(G_1^1) = \{\{X_1\}, \{Y_1\}, \{X_1, Y_1\}\}$, $FP(G_1^2) = \{\{X_1\}, \{Y_2\}, \{X_1, Y_2\}\}$, $FP(G_2^1) = \{\{X_1\}, \{Z_1\}, \{X_1, Z_1\}\}$, $FP(G_2^2) = \{\{Y_2\}, \{Z_2\}, \{X_2, Z_2\}\}$.

After filtering, $WP(G_1^1) = \{\{Y_1\}, \{X_1, Y_1\}\}, WP(G_1^2) = \{\{X_1, Y_2\}\}, WP(G_2^1) = \{\{Z_1\}, \{X_1, Z_1\}\}$, $WP(G_2^2) = \{\{Z_2\}, \{X_2, Z_2\}\}$

Identifying Author: Step 4 (Lines 26-35 in Algorithm 4.2) determines the author of the anonymous message ω by comparing ω with each writeprint $WP(G_i^g)$ of every suspect S_i and identifying the writeprint that is similar to ω. Intuitively, a writeprint $WP(G_i^g)$ is similar to ω if many frequent stylometric patterns in $WP(G_i^g)$ match the stylometric feature items found in ω.

The score function in Equation 4.2 is the modified form of Equation 4.1, which is used to measure the similarity between the anonymous message ω and a writeprint $WP(G_i^g)$. The proposed score function accumulates the support count of a frequent stylometric pattern.

$$Score(\omega \approx WP(G_i^g)) = \frac{\sum_{j=1}^{\rho} support(MP_j | G_i^g)}{|WP(G_i^g)|} \tag{4.2}$$

where $MP = \{MP_1, \cdots, MP_\rho\}$ is a set of matched patterns between $WP(G_i^g)$ and the anonymous message ω. As mentioned in Section 4.3.1, the higher the score means the higher similarity between the writeprint and the malicious message ω. The message ω is assigned to the writeprint of a message group G_a^g with the highest score. The suspect S_a of the group G_a^g is the plausible author of ω among the suspects.

4.4 Experiments and Discussion

The objectives of the experiments are: (1) to evaluate the two proposed methods, AuthorMiner1 and AuthorMiner2, in terms of authorship identification accuracy and to verify if the extracted writeprint exhibits strong evidence for supporting the conclusion on authorship attribution; (2) to measure the effect of the number of authors on the results; (3) to study the effect of the interval size and minimum support on the classification accuracy of AuthorMiner1; (4) to gauge the effects of the training size of a suspect on the conclusions; (5) to compare the accuracy score of the two methods with some previously developed classification methods.

In our experiments, we use 285 stylometric features including 99 lexical features, 158 syntactic features (150 function words and 8 punctuation marks), 15 structural features, and 13 domain-specific features. The features used in this study are discussed in Section 4.2.1. The function words used in our study are listed in Appendix I. Thirteen content-specific terms that are common across the Enron dataset are used.

We perform our experiments on the publicly available e-mail corpus, the Enron e-mail dataset[2], written by former Enron employees. After preprocessing, the corpus contains 200,399 real-life e-mails from 158 individuals [22]. To evaluate the authorship identification accuracy of our method, we randomly select n employees from the Enron e-mail dataset, representing n suspects $\{S_1, \cdots, S_n\}$. For each suspect S_i, we choose m of S_i's e-mails, where $\frac{2}{3}$ of the m e-mail messages are for training and the remaining $\frac{1}{3}$ of the m e-mail messages are for testing. Next, we apply AuthorMiner1, to extract

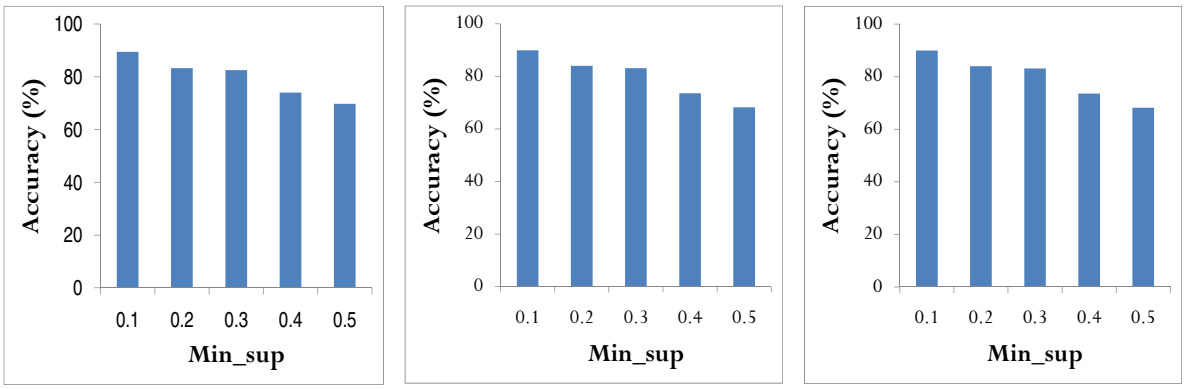
[2]http://www.cs.cmu.edu/∼enron/

the writeprints of $\{S_1, \cdots, S_n\}$ from the training set and then determine the author of each e-mail in the testing set. The authorship identification accuracy is measured by the percentage of correctly matched authors in the testing set.

The experimental results of the two approaches, AuthorMiner1 and AuthorMiner2, are discussed separately in the following two subsections.

4.4.1 AuthorMiner1

The purpose of experiments in this section is, to evaluate the presented approach of writeprint mining in authorship attribution from three main aspects. First, keeping the number of authors n and training size m constant, we study the effect of the number of discretized intervals and minimum support *min_sup* on the identification accuracy. Second, we measure the effect of the number of authors n on the classification score. Third, sample size m is another important parameter that needs to be evaluated in terms of authorship identification score.



(a) No. of discretized intervals = 2 (b) No. of discretized intervals = 4 (c) No. of discretized intervals = 6

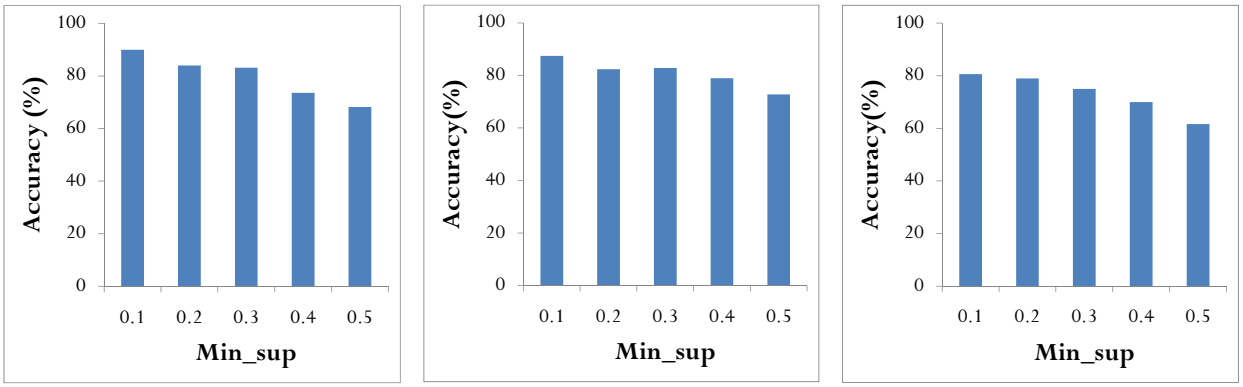
Figure 4.3: Accuracy vs. *Min_sup*, No. of discretized intervals (*Authors* = 6, *Messages* = 20)

In the first set of experiments, we consider six authors and selecting 20 messages for each author from Enron dataset. We discretize the normalized values of each feature into three intervals, i.e., 2, 4, and 6 and choosing the minimum support threshold *min_sup* = 0.1, 0.2, 0.3, 0.4, and 0.5. The experimental results are depicted in Figure 4.3. The accuracy spans from 67% to 89% at $min\_sup = 0.5$ through 0.1 (i.e., decrementing each successive value by 0.1), suggesting that our proposed method can effectively identify the author of an anonymous message based on the extracted writeprints when a reasonable *min_sup* is specified. As *min_sup* increases, the number of extracted frequent patterns, i.e., $|FP(M_i)|$, decreases and the extracted frequent patterns tend to capture the general writing style that is common to other suspects, thus, are likely to be eliminated by the filtering process of our method. As a result, the writeprint becomes less effective for authorship identification and the accuracy decreases.

In the effort to study the effect of the number of discretized intervals on the accuracy, we measure the authorship identification accuracy with respect to the number of intervals. We keep the number of authors n and the training size m constant. Figure 4.3 illustrates that the accuracy remains constant for different number of discretized intervals for a given *min_sup*, suggesting that our method is robust to the number of intervals.

Figure 4.4 depicts the effect of changing the number of authors on the authorship identification accuracy. We consider 6, 12, and 18 authors with 20 messages per author while keeping the interval size constant, i.e., 6. The accuracy drops from 89% for six authors to 80% for 18 authors. The accuracy drop is relatively small compared to the increase in the number of suspects. Most traditional classifiers usually have a significant

drop as the number of target classes (suspects) increases. These results suggest that our proposed method can effectively identify the author of a message when the candidate list of suspects is close to 20.



(a) No. of authors = 6 (b) No. of authors = 12 (c) No. of authors = 18

Figure 4.4: Accuracy vs. No. of authors ($Messages = 20$, No. of discretized intervals = 6)

The third set of experiments is designed to gauge the effect of sample size on the attribution accuracy while keeping all other parameters constant. By varying the number of messages per author m from 10 to 40, i.e., a multiple of 10, the accuracy spans from 87% to 89%, as shown in Figure 4.5. Though the change is not significant, however, it indicates that the accuracy increases by increasing the sample size of the suspects.

In addition to measuring the quality of writeprint using classification accuracy, we also manually examined the extracted writeprints and found that frequent patterns can succinctly capture combinations of features that occur frequently in a suspect's e-mails. Many of those hidden patterns are not obvious. Due to the fact that all the matched frequent patterns can be found in the anonymous (malicious) message, the frequent patterns themselves serve as a strong evidence for supporting the conclusion on authorship.

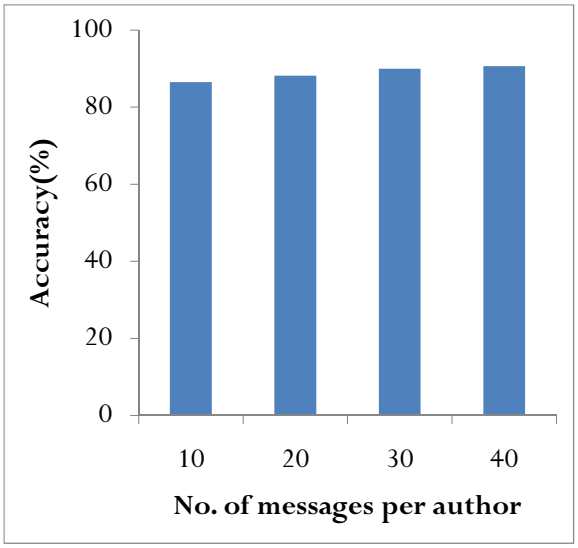


Figure 4.5: Accuracy vs. No. of messages per author ($Authors = 6$, No. of discretized intervals = 6, $Min\_sup = 0.1$)

4.4.2 AuthorMiner2

The objective of our experiments is, to evaluate the accuracy of AuthorMiner2 in authorship identification of anonymous messages. Next, we compare the accuracy of AuthorMiner2 with AuthorMiner1 as well as with few other authorship classification techniques. An identification is correct if the AuthorMiner2 or the traditional classification method can correctly identify the true author of an anonymous text message among the group of suspects. We employ 10-fold cross-validation to measure the authorship identification accuracy. The experiments are repeated for 4, 8, 12, 16, and 20 authors while keeping the training and testing set constant, i.e., 40 messages per author.

The experimental result of AuthorMiner2 for calculating authorship identification score is depicted in Figure 4.6. The accuracy drops from 92.37% to 71.19% by increasing the number of candidate authors from four to twenty. The accuracy of the proposed

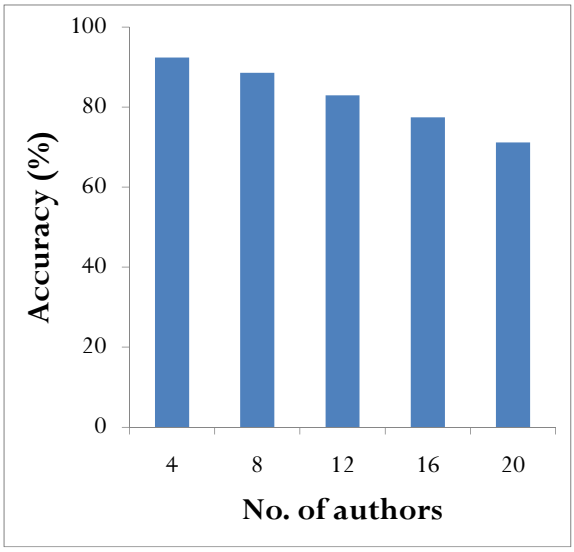


Figure 4.6: Experimental results of AuthorMiner2

approach increases by 2-8% as compared to AuthorMiner1. The authorship accuracy can be further improved by using a dataset that truly reflects the phenomenon of volatile stylistics. Because most of the e-mails in Enron dataset are official and are written to co-workers. The corpus does not contain messages written to friends and family members and of course it does not contain malicious e-mails.

Figure 4.7 depicts the average identification accuracy of AuthorMiner1, AuthorMiner2, and six classification methods namely Radial Basis Function Network (RBFNetwork) [18], Ensemble of Nested Dichotomies (END) [44], J48 [87], NaiveBayes [91], and BayesNet [82]. These methods are chosen because they are either popular in the field or the state-of-the-arts in their category. For example, RBFNetwork is an artificial neural network, J48 is a commonly employed decision tree classification method, and Naive Bayes is often used as a benchmark classifier for comparison. The selected classifiers are implemented in WEKA [111]

The identification accuracy is calculated for 4, 8, 12, 16, and 20 authors in all the methods. The newly proposed method, AuthorMiner2, outperforms all other techniques including AuthorMiner1. The two probability classifiers, i.e., Naive Bayes and BayesNet with accuracy score of 70% performed poorly for the given dataset. A similar accuracy trend can be seen in some previous studies including [121] and [50]. In some real-life investigation cases, the number of potential suspects is usually not very large.

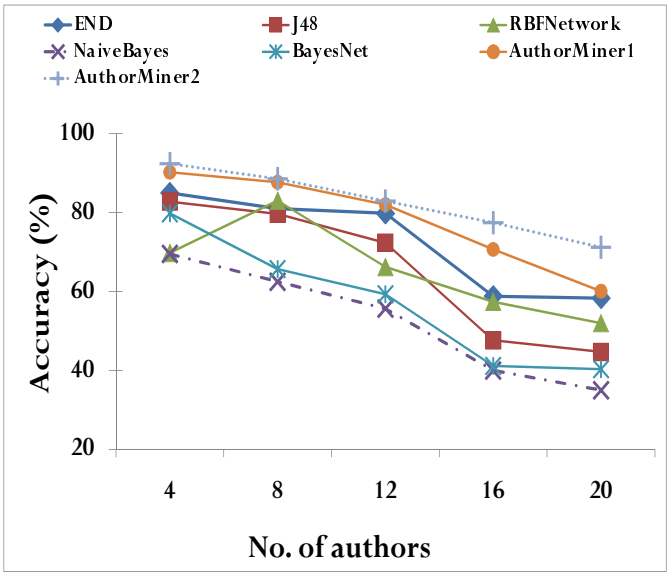


Figure 4.7: Comparing AuthorMiner2 with existing techniques

Figure 4.7 also suggests that the accuracy of AuthorMiner2 is relatively flat, as compared to other methods, implying that it is more robust to the change in the number of suspects. Furthermore, AuthorMiner2 can precisely model the sub-writeprints of a suspect in a presentable format. Other methods do not share this merit. The consistency in results of AuthorMiner1 and AuthorMiner2 indicates the robustness of our frequent-pattern-based writeprint mining for authorship identification.

AuthorMiner2 vs.	END	J48	RBFNetwork	NaiveBays	BaysNet	AuthorMiner1
test statistic value	2.966	3.242	5.555	8.552	5.207	2.848
reject H_0?	yes	yes	yes	yes	yes	yes

Table 4.6: Paired t test ($\alpha = 0.05$, $df = 4$, critical value $t_{0.05,4} = 2.132$)

The accuracy gap between AuthorMiner1 and AuthorMiner2 widens as the number of suspects increases. The improvement of AuthorMiner2 over AuthorMiner1 is contributed by the precise modeling of sub-writeprints.

To illustrate the statistical significance of the performance difference between AuthorMiner2 and other methods, we perform a paired t-test on the data in Figure 4.7 with the null hypothesis $H_0 : \mu_D = 0$ and the alternative hypothesis $H_a : \mu_D > 0$ where $\mu_D = \mu_{AuthorMiner2} - \mu_{other\_method}$. H_0 will be rejected if the test statistic value is greater than or equal to the critical value $t_{0.05,4} = 2.132$ at significance level 0.05. H_0 is rejected in all cases as shown in Table 4.6. The experimental result strongly suggests that the performance of AuthorMiner2 is better than the other six compared methods.

The choice of minimum support threshold *min_sup* affects the identification accuracy of our approach. Increasing the value of *min_sup* decreases the accuracy as the number of non-frequent patterns increases. The dropping such patterns we tend to loose at least some information. The accuracy score is relatively consistent by keeping *min_sup* between 0.1 to 0.3. The efficiency is inversely proportional to minimum support due to the increased number of frequent patterns.

The authorship identification process includes reading files, identifying writeprints,

and classifying an anonymous e-mail. The total runtime is dominated by the Apriori-based process of the frequent stylometric extraction in the writeprint identification process. Thus, the complexity of AuthorMiner1 and AuthorMiner2 is the same as the complexity of Apriori, which is $O(|U|^l \times |M_i|)$, where $|U|$ is the number of distinct stylometric feature items, l is the maximum number of stylometric features of any e-mail, and $|M_i|$ is the number of training samples from suspect S_i. In practice, l usually peaks at 2 [55]. For any test case of AuthorMiner2 shown in Figure 4.7, the total runtime is less than 7 minutes.

In addition to identification accuracy, AuthorMiner1 and AuthorMiner2 can precisely model the writeprint of a suspect in a presentable format. For example, the writeprint of an author called *fossum-d* consists of 86 frequent stylometric patterns. We show two of them below:

{*f91*:low, *f92*:low} with $support = 23$

{*f243*:high, *f244*:high} with $support = 18$

where *f91* measures the ratio of the number of distinct words and total words, *f92* measures the vocabulary richness using hapax legomena, *f243* measures the frequency of the function word "where", and *f244* measures the frequency of the function word "whether". These two patterns imply that *fossum-d*'s vocabulary richness is low and *fossum-d* often uses the words "where" and "whether" in his/her e-mails.

4.5 Summary

In this chapter, we have defined two authorship identification problems. First, attribution of an anonymous message to the true author by ignoring the occasional stylistic variation of the potential authors. Second, attribution of an anonymous message with contextual stylistic change of potential suspects. The first problem is further refined into three sub-problems: (1) extracting the writeprint of a suspect; (2) identifying the author of a malicious e-mail; and (3) collecting evidence for supporting the conclusion on authorship. Generally, the same methodology is applied in the court of law for resolving the attribution issues. Most previous contributions focused on improving the classification accuracy of authorship identification, but only few of them studied how to gather strong evidence for the court of law.

To address the first problem, we introduce a novel approach of authorship attribution and formulate a new notion of writeprint based on the concept of frequent patterns. Unlike the writeprints in previous literature that are a set of predefined features, our notion of writeprint is dynamically extracted from the data as combinations of features that occur frequently in a suspect's messages, but not frequently in other suspect's messages. The experimental results on real-life e-mail dataset suggest that the identified writeprint does not only help identifying the author of anonymous e-mail, but also presents intuitive yet strong evidence for supporting the authorship finding. Due to its intuitiveness, non-technical personnel including the judge and jury in a law court can understand it.

To address the second problem, we extend and improve our approach of frequent pattern-based writeprint to capture the sub-styles of a suspect by creating sub-writeprints

of a suspect. Comparing the accuracy score of AuthorMiner2 with AuthorMiner1 and some other techniques, suggests that by focusing on the sub-stylistics of an author prior to applying attribution methods increases the accuracy of the system.

This novel approach opens up a new promising direction of authorship attribution. We will further extend our tool to adopt different types of stylometric features and utilize the concept of frequent patterns to identify hidden writeprint of individuals for the purpose of messaging forensics. Similarly, more interesting results can be obtained by using the proposed approach on real e-mail traffic containing malicious messages.

Chapter 5

Authorship Attribution with Few Training Samples

The problem defined in this chapter is different in two aspects from the traditional authorship identification problem, discussed in the previous chapter of this thesis. First, the traditional authorship attribution studies [34, 121] assume to have large training samples of each candidate author, enough to build a classification model. In the current problem, we assume to have *few* training samples for each suspect. In some scenarios no training samples may exist and the suspects may be asked (usually through court orders) to produce a writing sample for investigation purposes. Second, in traditional authorship studies the problem is to attribute a *single* anonymous document to its true author. In the current study we assume to have more than one anonymous messages that need to be attributed to the true author(s). It is likely that the perpetrator may either create a ghost e-mail account or hack an existing account and then use it for sending illegitimate messages.

To address the aforementioned shortfalls, we redefine the authorship attribution problem as follows: given a collection of anonymous messages potentially written by a set of suspects $\{S_1, \cdots, S_n\}$, a cybercrime investigator first wants to identify the major groups of messages based on stylometric features; intuitively, each message group is written by one suspect. Then s/he wants to identify the author of each anonymous message collection from the given candidate suspects.

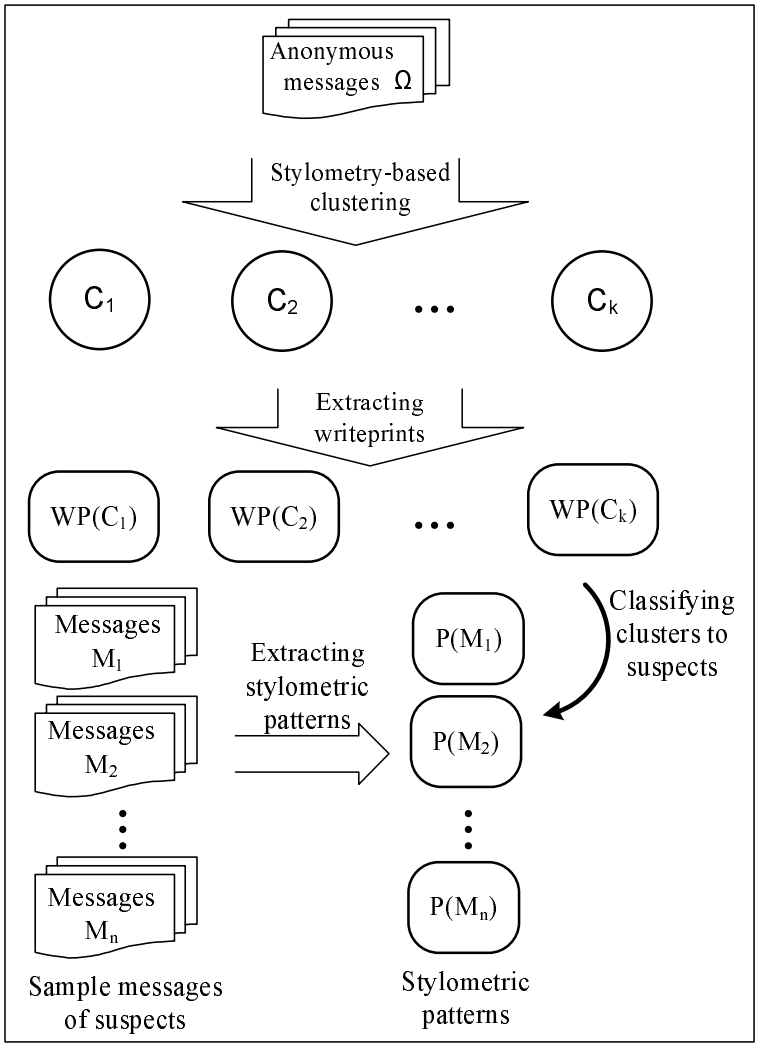


Figure 5.1: AuthorMinerSmall: Authorship identification with small training samples

We extend our stylometric pattern-based approach of AuthorMiner1 (described in Chapter 4), called *AuthorMinerSmall*, to address the newly defined problem. First, we extract the stylometric features from the given anonymous message collection Ω. As described in Section 2.3.1, the stylometric features include lexical features, style markers (punctuations and function words), structural features, and content-specific features. Each message is converted into a feature vector using vector space model representation. Then, we apply stylometry-based clustering to cluster the given messages into different groups. The intuition is that clustering by stylometric features can group the messages of the same author together. The subsequent steps of the proposed method are applicable only if this hypothesis is true. Our experimental results support the hypothesis.

Note that clustering applied in this chapter is different from traditional text clustering [46, 66] in two ways. First, the objective of the traditional clustering is to identify the different topics contained in the documents in question. The purpose of clustering in our context is to identify pertinent writing styles in the messages. Second, traditional clustering is applied on the basis of content-specific words, while in our case clustering is applied on the basis of stylometric features.

Messages of each cluster are used to extract the frequent stylometric patterns by applying our first approach, AuthorMiner1, described in Chapter 4. To compute the writeprint of a cluster $C_i \in \{C_1, \cdots, C_k\}$, the stylometric patterns shared by more than one cluster are deleted. Next, the stylometric patterns $\{P(M_1), \cdots, P(M_k)\}$ from the training samples $\{M_1, \cdots, M_k\}$ of the suspects are extracted. Finally, we compare each writeprint $WP(C_j)$ with every pattern $P(M_i) \in \{P(M_1), \cdots, P(M_k)\}$ to identify the most conceivable

author S_a of cluster C_j.

Cluster analysis provides the crime investigator a deep insight on the writing styles found in the given anonymous e-mails, in which the clusters and the extracted writeprint could serve as input information for higher-level data mining. To investigate the relative discriminating power of different stylometric features, clustering is applied to each feature type (i.e., lexical, syntactic, structural, and content-specific) separately. In our experiments, we gauge the effects of the number of authors and the size of training set on the purity of clusters. Using visualization and browsing features of our developed tool, an investigator can explore the process of cluster formation and evaluation.

We summarize the contributions of this research work as follows:

- *Attribution based on few training samples:* Existing authorship identification methods often require a reasonably large number of training samples in order to build a classification model. Our proposed method is effective even if only a few training samples exist.

- *Clustering by stylometric features:* In the data mining community, content-based clustering is used to cluster messages into different groups based on the topic. Employing the same notion, our experimental results on a real-life e-mail corpus (Enron e-mail corpus [22]) suggest that clustering by stylometric features is a sensible method to group together messages written by the same person.

- *Cluster analysis:* We propose a method and develop a tool for an investigator to visualize, browse, and explore the writing styles extracted from a collection of anonymous e-mails. The relative strength of different clustering algorithms is evaluated.

Our study reveals the relative discriminating power of four different categories of stylometric features. We study the effects of the number of suspects as well as the number of messages per suspect on the clustering accuracy.

- *Dataset attribution:* Our proposed method can be used to attribute a collection of messages (e.g., anonymous message or a ghost e-mail account) to its plausible author.

The remaining of the chapter is organized as follows: Section 5.1 defines the problem statement. Section 5.2 presents the approach of dataset attribution based on small training data. Section 5.3 examines the viability of the proposed approach based on experiments on real-life dataset. Section 5.4 concludes the chapter.

5.1 Problem Statement

The problem of *authorship attribution with few training samples* is to identify the most plausible author S_a of a set of anonymous text messages Ω from a group of suspects $\{S_1, \cdots, S_n\}$, with only *few* sample text messages M_i for each suspect S_i. Note, this problem is different from the first problem in Definition 4.1.1: (1) The number of training samples $|M_i|$ is small (say less than 30 sample e-mails). Therefore, it is infeasible to build a classifier as in the traditional classification method [60, 121] or to extract the *frequent* stylometric patterns based on low support counts. (2) The first problem focuses on how to identify the author of one anonymous message. In contrast, this problem focuses on how to cluster the anonymous text messages by stylometric features such that the messages of

each cluster are written by the same author, and how to identify the author of each cluster of anonymous messages. The investigator needs to support his findings with convincing evidence. The problem is formally described as follows.

Definition 5.1.1 (Authorship attribution with few training samples)**.** Let Ω be a set of anonymous text messages. Let $\{S_1, \cdots, S_n\}$ be a set of suspected authors of Ω. Let $\{M_1, \cdots, M_n\}$ be the sets of text messages previously written by suspects $\{S_1, \cdots, S_n\}$, respectively. Assume $|M_i|$ is very small. The *problem* is to first group the messages Ω into clusters $\{C_1, \cdots, C_k\}$ by stylometric features, and then to identify the plausible author S_a from $\{S_1, \cdots, S_n\}$ for each cluster of anonymous messages $C_j \in \{C_1, \cdots, C_k\}$, with presentable evidence. The most plausible author S_a of C_j is the suspect whose stylometric patterns $P(M_i)$ have the "best match" with writeprint WP(C_j). ■

5.2 Proposed Approach

The general idea of our proposed method, depicted in Figure 5.1, is composed of five steps. Step 1 involves the preprocessing, feature extraction, and normalization. Step 2 is grouping anonymous messages Ω into clusters $\{C_1, \cdots, C_k\}$ by stylometric features such that each cluster contains the anonymous messages written by the same suspect. Step 3 is feature discretization and frequent stylometric patterns mining from each cluster of messages. Step 4 is calculating the writeprint of each cluster by filtering the frequent stylometric patterns shared by two or more clusters. Step 5 is identifying the most plausible author S_a of each cluster C_j by comparing the extracted writeprint $WP(C_j)$ with every set of training samples $M_i \in \{M_1, \cdots, M_n\}$.

5.2.1 Preprocessing

The preprocessing applied in this section is different from the preprocessing applied in the previous chapter. In this chapter, we do not apply discretization after the usual process of cleaning, tokenization, stemming, and feature extraction. Discretization is applied after the clusters are formed in the next section. Similarly, the preprocessing step of stylometry-based clustering [58] is different from the traditional text clustering [69]. In the traditional text clustering only the content-specific words are counted while in stylometry-based clustering, the stylometric features are extracted in addition to the content-specific words.

Using vector space model representation, each message μ is converted into a 285-dimensional vector of features $\mu = \{X_i, Y_j, Z_k\}$, as shown in Table 5.2. When all messages are converted into feature vectors, normalization is applied to the columns as needed. Discretization of the extracted features $\{X, Y, Z\}$ into respective *feature items* is done after the clustering phase.

5.2.2 Clustering by Stylometric Features

Clustering groups the anonymous messages Ω into different clusters $\{C_1, \ldots, C_k\}$ on the basis of stylometric features. The hypothesis is that the writing style of every suspect is different, so clustering by stylometric features could group the messages written by the same author into one cluster. The experimental results in our previous work [59] support the hypothesis. This clustering step is very different from AuthorMiner1 in Section 4.3.1, which groups *training samples* with the goal of identifying the sub-writeprints of a suspect. In contrast, the reason of clustering *anonymous messages* in AuthorMinerSmall is to

facilitate more precise writeprint extraction, which is otherwise impossible due to small training data.

One can apply any clustering methods, such as k-means, to group the anonymous messages into clusters $\{C_1, \ldots, C_k\}$ such that messages in the same cluster have similar stylometric features and messages in different clusters have different stylometric features. Often, k is an input parameter to a clustering algorithm. In this case, k can be the number of suspects.

We evaluate our proposed method by employing three clustering algorithms: Expectation Maximization (EM), k-means, and bisecting k-means. We choose the k-means clustering algorithm [53] because it is known to be both simple and effective. The k-means algorithm partitions a set of objects into k sub-classes. It attempts to find the centers of natural clusters in the data by assuming that the object attributes form a vector space, and minimizing the intra-cluster variance. Thus, k-means generally forms, circular clusters around a centroid, and the algorithm outputs the centroids. k-means is particularly applicable to numeric attributes. *Expectation Maximization (EM) algorithm*, first proposed in [37], is often employed where it is hard to predict the value of k (number of clusters). For instance, during forensic analysis of anonymous e-mails, an investigator may not know the total number of suspects (or different writing styles) within a collection. In a more common scenario, a user may want to validate the results obtained by other clustering algorithms say k-means, or bisecting k-means.

Suppose we have 12 anonymous messages and after applying clustering we obtain three clusters denoted by $\{C_1, C_2, C_3\}$ as shown in Table 5.1.

Table 5.1: Clusters with member messages

Cluster C	**Message** (μ)	**Feature values**
C_1	μ_1	{0.130,0.580,0.555}
C_1	μ_2	{0.132,0.010,0.001}
C_1	μ_3	{0.133,0.0124,0.123}
C_2	μ_4	{0.119,0.250,0.345}
C_2	μ_5	{0.0,0.236,0.532}
C_2	μ_6	{0.150,0.570,0.679}
C_3	μ_7	{0.0,0.022,0.673}
C_3	μ_8	{0.985,0.883,0.990}
C_3	μ_9	{0.137,0.444,0.894}
C_3	μ_{10}	{0.0,0.455,1.000}
C_3	μ_{11}	{0.134,0.012,0.0}
C_3	μ_{12}	{0.0,0.123,1.000}

To measure the purity of clusters and validate our experimental results, we use the F-measure [46]. F-measure is derived from *precision* and *recall*, the accuracy measures commonly employed in the field of information retrieval. The aforementioned three functions are shown by the following mathematical equations.

$$recall(N_p, C_q) = \frac{O_{pq}}{|N_p|} \tag{5.1}$$

$$precision(N_p, C_q) = \frac{O_{pq}}{|C_q|} \tag{5.2}$$

$$F(N_p, C_q) = \frac{2 * recall(N_p, C_q) * precision(N_p, C_q)}{recall(N_p, C_q) + precision(N_p, C_q)} \tag{5.3}$$

where O_{pq} is the number of members of actual (natural) class N_p in cluster C_q, N_p is the actual class of a data object O_{pq} and C_q is the assigned cluster of O_{pq}.

5.2.3 Frequent Stylometric Pattern Mining

Once clusters $\{C_1, \cdots, C_k\}$ are formed, the next step is to calculate the writeprint of each cluster $C_i \in \{C_1, \cdots, C_k\}$. The pattern mining helps unveil the hidden association between different stylometric features. By feature items, we mean the discretized value of a feature, which is discussed in the following paragraph. We capture such frequently occurred patterns by the concept of *frequent itemset* [7], in a way similar to the one described in [60] and Chapter 4 of this thesis.

Table 5.2: Clustered messages after discretization

Cluster C	**Message** (μ)	**Stylometric Features**
C_1	μ_1	$\{X_2, Y_2, Z_2\}$
C_1	μ_2	$\{X_2, Y_1, Z_1\}$
C_1	μ_3	$\{X_2, Y_1, Z_1\}$
C_1	μ_4	$\{X_1, Y_1, Z_1\}$
C_2	μ_5	$\{Y_1, Z_2\}$
C_2	μ_6	$\{X_3, Y_2, Z_2\}$
C_2	μ_7	$\{Y_1, Z_2\}$
C_2	μ_8	$\{X_3, Y_2, Z_2\}$
C_3	μ_9	$\{X_2, Y_1, Z_3\}$
C_3	μ_{10}	$\{Y_1, Z_3\}$
C_3	μ_{11}	$\{X_2, Y_1\}$
C_3	μ_{12}	$\{Y_1, Z_3\}$

To extract frequent stylometric patterns from each cluster, we apply Apriori algorithm [7]. The Apriori algorithm can not be applied to numeric data. Therefore, we need to split feature values into appropriate intervals. For this, we discretize each normalized frequency of a feature $F_a \in \{F_1, \cdots, F_g\}$ into a set of intervals $\{\imath_1, \cdots, \imath_h\}$, called *feature items*. Detailed description of our proposed discretization method is given in Section 4.2.2. Table 5.2 shows the discretized form of the messages shown in Table 5.1.

Table 5.3: Frequent stylometric patterns for clusters C_1, C_2, C_3

Cluster(C)	Frequent Stylometric Patterns (FP)
C_1	$\{X_2\},\{Y_1\},\{Z_1\},\{X_2,Y_1\},\{X_2,Z_1\},\{Y_1,Z_1\},\{X_2,Y_1,Z_1\}$
C_2	$\{X_3\},\{Y_1\},\{Y_2\},\{Z_2\},\{X_3,Y_2\},\{X_3,Z_2\},\{Y_1,Z_2\},\{Y_2,Z_2\},\{X_3,Y_2,Z_2\}$
C_3	$\{X_2\},\{Y_1\},\{Z_3\},\{X_2,Y_1\},\{Y_1,Z_3\}$

Detailed description of extracting frequent stylometric patterns from e-mail messages, is given in Section 4.3.1.

We use a running example to explain the proposed approach of writing style mining. Suppose we have three clusters, C_1 with messages $\{\mu_1,\mu_2,\mu_3,\mu_4\}$, C_2 with messages $\{\mu_5,\mu_6,\mu_7,\mu_8\}$, and C_3 containing messages $\{\mu_9,\mu_{10},\mu_{11},\mu_{12}\}$, as shown in Table 5.2. To calculate frequent stylometric patterns for each cluster, we assume that the user-defined $min\_sup = 0.5$. It means that a pattern P is frequent in C_i if at least 2 out of 4 e-mails (by truncating the decimal part) within a cluster C_i contain all feature items in P. The frequent stylometric patterns associated with each cluster are shown in Table 5.3. For instance, pattern $\{X_2,Y_1,Z_1\}$ is a frequent pattern in C_1 because at least 2 out of 4 e-mails of cluster C_1 contain this pattern. The lists of frequent stylometric patterns are shown in Table 5.3.

5.2.4 Writeprint Mining

A writeprint is the disjoint set of frequent stylometric features. Therefore, the patterns that are shared by more than one clusters are dropped. For instance, in our example, $\{X_2\}$ and $\{X_2,Y_1\}$ are shared by cluster C_1 and C_3, $\{Y_1\}$ is shared by all the three clusters. $\{X_2,Y_1\}$ is shared by C_1 and C_3. Therefore, these patterns are deleted. The remaining

Table 5.4: Writeprints for clusters C_1, C_2, C_3

$WP(C_1)$	$\{X_2, Z_1\}, \{Y_1, Z_1\}, \{X_2, Y_1, Z_1\}$
$WP(C_2)$	$\{X_3\}, \{X_3, Y_2\}, \{X_3, Z_2\}, \{Y_2, Z_2\}, \{X_3, Y_2, Z_2\}$
$WP(C_3)$	$\{Z_3\}, \{Y_1, Z_3\}$

frequent patterns constitute the unique writeprints $WP(C_1), WP(C_2), WP(C_3)$, as shown in Table 5.4.

5.2.5 Identifying Author

In this section, we identify the most plausible author for each cluster of anonymous messages C_j by comparing $WP(C_j)$ with the training samples $\{M_1, \cdots, M_n\}$. For each message in M_i, we extract the stylometric feature items, denoted by $\{P(M_1), \cdots, P(M_n)\}$. If there are two or more samples, we take the average of the feature items over all the messages in M_i. The similarity between C_i and M_i is computed by using Equation 5.4. The most plausible author is the suspect having the highest score.

$$Score(M_i \approx WP(C_i)) = \frac{\sum_{j=1}^{p} support(MP_j|C_i)}{|WP(C_i)|} \tag{5.4}$$

where $MP = \{MP_1, \cdots, MP_p\}$ is a set of matched patterns between $WP(C_i)$ and the message sample M_i of suspect S_i. The score is a real number within the range of $[0, 1]$. The higher the score means the higher the similarity between the cluster writeprint $WP(C_i)$ and the message sample M_i. The author of message sample M_i having the highest score for a cluster is the true author of that cluster.

Suppose a message M_1 contains two patterns $\{X_3\}$ and $\{Y_1, Z_2\}$. Suppose the support of $\{X_3\}$ is 2 in cluster C_2 and the support of $\{Y_1, Z_2\}$ in cluster C_3 is 4. Using

Equation 5.4 the score of cluster C_2 for M_1 is 0.4 and that of cluster C_3 is 4. Therefore, cluster C_3 is attributed to suspect S_1. The same process is repeated for the remaining two clusters as well.

In an unlikely case where multiple suspects have the same highest score for a given cluster, the strategy discussed in Section 4.3.1 is applied.

5.3 Experiments and Discussion

The objective of our experiments is to evaluate the authorship identification accuracy of the proposed approach *AuthorMinerSmall*. For this, first we show that clustering by stylometric features can be employed to group together the messages of an author. This is a two step process. First, we cluster the randomly selected messages. Second, we use F-measure [67] to measure the similarity between the cluster solution and the true author labels. The higher the F-measure implies the better the cluster quality. F-measure has a range [0,1].

The cluster analysis experiments help answer the following questions. Which of the clustering algorithm perform better than others for a given message dataset? What is the relative strength of each of the four different types of writing style features? What is the effect of changing the number of authors on the experimental results? What is the effect of changing the number of messages per author on the cluster quality.

To find the answer of first question, we employ three clustering algorithms, namely Expectation-Maximization (EM), k-means, and bisecting k-means. The cluster quality of the three algorithms is measured while all other parameters, e.g., stylometric features,

number of authors, and size of training data are kept constant. To answer the second question, we apply clustering over 15 different combinations of stylometric features. Next, we change the number of authors while keeping other parameters, e.g., feature set and size of training samples, constant. In the fourth set of experiments, we check the effects of changing the number of messages per author on the clustering result.

We use a real-life dataset, Enron e-mails [22], which contains 200,399 e-mails of about 150 employees of Enron corporation (after cleaning). We randomly selected h employees from the Enron e-mail dataset, representing h authors $\{A_1, \cdots, A_h\}$. For each author A_i, we select x of A_i's e-mails; where h varies from three to ten while value of x is selected from {10, 20, 40, 80, 100}.

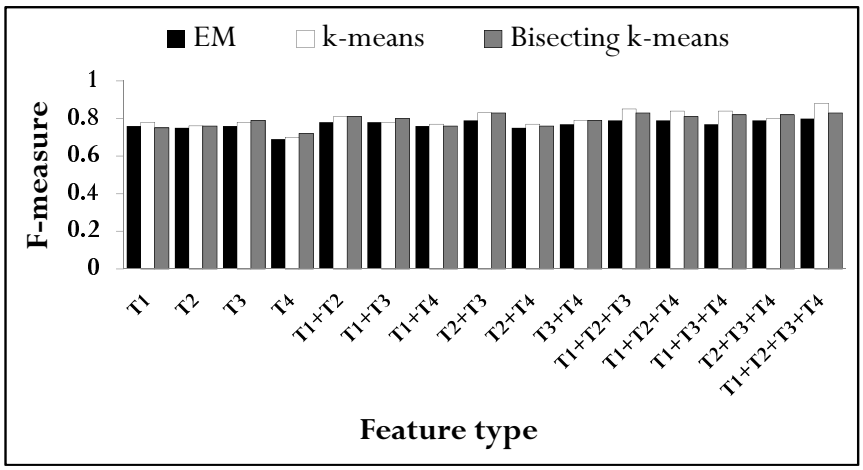


Figure 5.2: F-measure vs. Feature type ($Authors = 5$, $Messages = 40$)

In the first set of experiments, we select 40 e-mails from each one of the five authors. Results of the three clustering algorithms are shown in Figure 5.2. The value of F-measure spans from 0.73 to 0.80 for EM, from 0.73 to 0.88 for k-means, and from 0.75 to 0.83 for bisecting k-means. The better result of k-means and bisecting k-means over EM (at

least in this set of experiments) indicates that knowing the number of clusters k, one can obtain better results. Result of k-means is better than bisecting k-means. Initially these results seemed unexpected which were later on validated after completing all sets of experiments. k-means performed better as compared to bisecting k-means for up to 40 e-mails per author. By increasing the number of e-mails per author beyond 40, the accuracy of bisecting k-means starts increasing. It suggests that bisecting k-means is more scalable than EM and k-means.

The experimental results of Figure 5.2, help measure the discriminating power of the different stylometric features. For this, we use 15 possible combinations of these features. Looking at the individual features, content-specific features (denoted by T_4) perform poorly while style markers (denoted by T_2) and structural features (denoted by T_3) produce best clustering results. These two trends are matching with the previous stylometric studies [34, 121]. Over all, the best results are obtained by applying k-means on $T_1+T_2+T_3+T_4$, i.e., the combination of all four types of features. By adding contents-specific features to $T_1+T_2+T_3$, we do not see any noticeable improvement in the accuracy of EM and bisecting k-means. The selected keywords are probably common among e-mails of the selected authors. Another important observation is that the performance of T_2+T_3 is better than any other combination of two feature, e.g., T_1+T_2 and T_1+T_3.

In the second set of experiments, we consider use all the four types of stylometric features, i.e., $T_1+T_2+T_3+T_4$ and select 40 messages per author. As depicted in Figure 5.4, the experiments are repeated for 4, 8, 12, 16, and 20 authors. The value of F-measure reaches 0.91 for four authors using bisecting k-means. Accuracy of the three

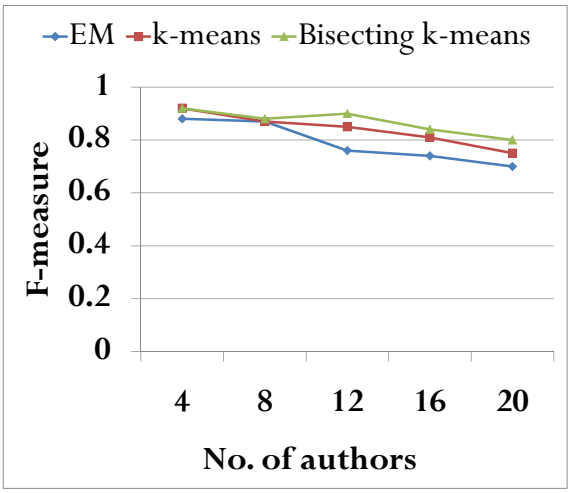


Figure 5.3: F-measure vs. No. of authors ($Messages = 40$, $Features = T_1 + T_2 + T_3 + T_4$)

clustering algorithms drops as more authors are added to the experiments.

In the next set of experiments, we evaluate the effects of the training size by keeping the number of authors (five) and feature set ($T_1 + T_2 + T_3 + T_4$) unchanged. As depicted in Figure 5.4, the value of F-measure increases by increasing the number of messages per author. k-means and bisecting k-means achieve 90% purity for 40 messages per author while the results of EM are not consistent. Increasing the number of messages per author beyond 40 negatively affect results of all the three algorithms. Among the three algorithms, the accuracy of EM drops faster than the other two, and bisecting k-means is more robust compared to simple k-means. These results explain the relative behavior of these algorithms in terms of scalability.

The best accuracy is achieved by applying k-means over a combination of all four feature types when e-mails per user is limited to 40. Bisecting k-means is a better choice when there are more authors and the training set is larger. By taking into account the topic of discussion, better results can be obtained by selecting domain-specific words carefully.

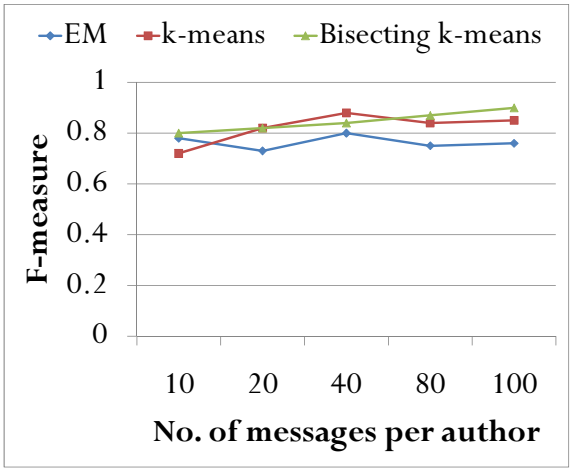


Figure 5.4: F-measure vs. No. of messages per author ($Authors = 5$, $Features = T_1 + T_2 + T_3 + T_4$)

One way could be to identify author-specific keywords by apply content-based clustering on e-mails of each author separately. Results of *EM* are insignificant and are hard to improve by parameter tuning.

Next, we evaluate the authorship identification accuracy of AuthorMinerSmall. We randomly select 40 text messages from each suspect. Selecting 36 out of the 40 messages from each suspect for training while the remaining 4 messages from each suspect are used for testing. Let n be the number of suspects. Then, we cluster the $36 \times n$ messages by stylometric features using k-means, and then match each cluster of anonymous messages with the remaining $4 \times n$ messages with known authors. An identification is correct if AuthorMinerSmall can correctly identify the true author of an anonymous text message among the group of suspects.

Figure 5.5 depicts the authorship identification accuracy for AuthorMinerSmall with the number of suspects ranging from 4 to 20. When the number of suspects is 4, the accuracy is 81.18%. When the number of suspects increases to 20, the accuracy drops

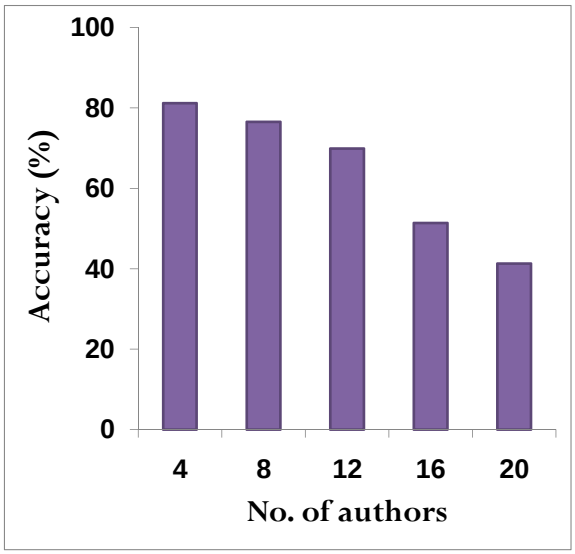


Figure 5.5: AuthorMinerSmall: Accuracy vs. No. of authors

to 41.26%. Given that the training dataset is so small, the accuracy above 70% is in fact very encouraging when the number of suspects is not too large.

The computational complexity of the AuthorMinerSmall is based on two phases. (1) The computational complexity of clustering phase depends on the clustering algorithm. For instance, it is $O(k \times |\Omega|)$ for k-means, where k is the number of clusters and $|\Omega|$ is the number of anonymous messages. (2) The computational complexity of writeprint extraction phase is $O(|U|^l \times |C_i| \times k)$, where $|U|$ is the number of distinct stylometric feature items, l is the maximum number of stylometric features of any e-mail, and $|C_i|$ is the number of anonymous messages in cluster C_i. As discussed in Section 4.4.2, l usually peaks at 2. For any test case of AuthorMinerSmall shown in Figure 5.5, the total runtime is less than a minute.

5.4 Summary

The non-availability of enough training samples of potential suspects is one of main limitation of the criminal investigation process. To address this issue, we have presented a method for authorship identification of anonymous messages based on few training samples. The approach is primarily based on the intuition that clustering by stylometric features is a sensible method to divide text messages into different groups. We argue that the hypothesis is true based on our experiments on real-life e-mails. Moreover, we show that using cluster analysis, an investigator can get a deeper insight of the anonymous messages and learn about the potential perpetrators. The writing styles in terms of feature patterns provide more concrete evidence than producing some statistical numbers.

The identification accuracy of AuthorMinerSmall for up to ten suspects is high while the accuracy above ten authors is low, which can be improved by tuning the parameters. For instance, selecting large size e-mails, increasing the number of stylometric features, and using sophisticated distance functions can help improve the accuracy score of the presented approach.

The current study suggests that content-specific keywords can be more effectively used for authorship identification in specific contexts, e.g., cybercrime investigation. The need is to develop robust techniques for selecting more appropriate words from the given suspicious dataset. Another important research direction would be to identify optimized set of stylometric features applicable in all domains. Most often contents of the same message are written in more than one language. Therefore, addressing the issue of language multiplicity is important especially for cybercrime investigation.

Chapter 6

Authorship Characterization

The problem of *authorship characterization* is to determine the sociolinguistic characteristics of the potential author of a given anonymous text message. Unlike the problems of authorship attribution, where the potential suspects and their training samples are accessible for investigation, no candidate list of suspects is available in authorship characterization. Instead, the investigator is given one or more anonymous documents and is asked to identify the sociolinguistic characteristics of the potential author of the documents in question. Sociolinguistic characteristics include ethnicity, age, gender, level of education, and religion [105].

In this chapter, we consider the worst case scenario of authorship characterization, in which even the data from the sample population is not enough to build a classifier. Our proposed approach, depicted in Figure 6.1, first applies stylometry-based clustering on the given anonymous messages Ω to identify major stylistic groups. The intuition is that clustering by stylometric features can cluster messages of an author in one group.

This intuition is supported by experimental results in Section 5.3. A group of messages is denoted by $C_i \in \{C_1, \cdots, C_n\}$. Next, we train a model on messages collected from the sample population. The developed model is employed to infer the characteristic of a potential

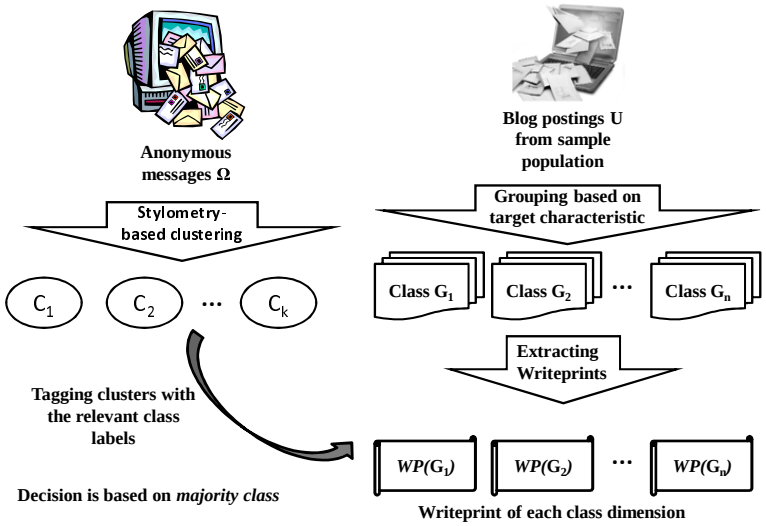


Figure 6.1: *AuthorCharacterizer*: Inferring characteristics of anonymous author

We use a blog dataset in our experiments as most bloggers voluntarily post their personal characteristics on their blogs. The selected bloggers need to be from the same class category that we want to infer in our experiments. For instance, to infer the gender of an author, we need to collect blog postings of male and female bloggers. Next, we precisely model the *writeprint* of each class category of the sample population by employing the concept of *frequent patterns* [7], detailed in Chapter 4. The extracted writeprints are then used to identify the class label of each message in a cluster C_i. We apply our approach for predicting two characteristics: gender and region or location. In the remainder of this chapter, we use the term "online messages" to indicate e-mail messages, blog postings,

and chat logs.

The contributions of this study are summarized as follows:

- *Characterization by frequent pattern-based writeprint:* In traditional authorship studies the characterization problem is addressed mostly by employing classifiers. This is the first work to use frequent pattern-based writeprint to infer an author's characteristics. The writeprint, the combination of co-occurring stylistic features, helps manifest the hidden association between the stylometric features.

- *Preliminary information:* Often, an investigator is provided with only a collection of anonymous suspicious messages and is asked to collect forensically relevant evidence from them. Clustering by stylometric features can be used to initiate the investigation process by identifying groups of stylistics; each group, intuitively, represents one suspect.

- *Small sample population:* Our frequent pattern-based approach can be used even if the size of the sample population is small or their sample messages are small. Koppel et al. [64] used blogs of approximately 47,000 bloggers, posted for one year, much larger than the dataset used in our experiments.

- *New category of characteristic:* Existing studies have investigated characteristics such as age, gender, educational level, and language background. We introduce a new dimension of authorship profiling, called *region or location*. Our experiments on blog postings collected from bloggers in Australia, Canada, and the United Kingdom suggest that the proposed method can be employed to predict (with certain

accuracy) a suspect's region.

The rest of the chapter is organized as follows. Section 6.1 formally defines the problem of authorship characterization. The proposed approach for addressing the characterization problem is described in Section 6.2. The approach is evaluated by experimentation on real-life data in Section 6.3. Summary of the chapter is given is Section 6.4.

6.1 Problem Statement

The problem of authorship characterization is defined as follows: Given a collection of anonymous online messages potentially written by some suspects, the task of an investigator is to identify the cultural and demographic characteristics of each suspect. We assume to have no candidate list of potential suspects and of course no training data from the suspects. The investigator assumes to have access to some online messages with known authors who come from the population of the suspects. The sample messages can be collected from blog postings and social networks that explicitly disclose the authors' public profiles.

Definition 6.1.1 (Authorship characterization)**.** Let Ω be a set of anonymous text messages potentially written by some suspects. The number of suspects may or may not be known a priori. Both scenarios are addressed in this study. Let U be a set of online text documents, collected from the same population of suspects, with known authors' characteristics. The *problem of authorship characterization* is to first group the messages Ω into clusters $\{C_1, \ldots, C_k\}$ by stylometric features, then identify the characteristics of

the author of each cluster C_j by matching the writeprint extracted from the online text documents U. ■

6.2 Proposed Approach

To address the authorship problem in Definition 6.1.1, we propose a method, called *AuthorCharacterizer*, to characterize the properties of an unknown author of some anonymous messages. Figure 6.1 shows an overview of AuthorCharacterizer in three steps. Step 1 is identifying the major groups of stylometric features from a given set of anonymous messages Ω. Step 2 is extracting the writeprints for different categories of online users from the given sample documents U. Step 3 is characterizing the unknown authors of Ω by comparing the writeprints with Ω.

6.2.1 Clustering Anonymous Messages

Once all anonymous messages contained in Ω are converted into feature vectors using vector space model representation technique, the next step is to apply clustering. For clustering, we have selected Expectation Maximization (EM), k-means, and bisecting k-means clustering algorithms. Because, k-means is more commonly used than other methods while *EM* is the preferred choice if the number of clusters (the number of suspects in our case) is not known a priori. Bisecting k-means performs better than k-means in terms of accuracy. Clustering is applied on the basis of stylometric features, the way similar to Section 5.2.2, which results in a set of clusters $\{C_1, \ldots, C_k\}$. The only difference is that the number of clusters k is the number of categories identified for a characteristic. For

instance, $k = 2$ (male/female) for gender, $k = 3$ (Australia/Canada/United Kingdom) for region or location.

6.2.2 Extracting Writeprints from Sample Messages

In our study, we use the blog postings collected from `blogger.com` because this website allows bloggers to explicitly mention their personal information. Each blog posting is converted into a set of stylometric feature items. Then we group them by the characteristics that we want to make inferences on the anonymous messages C_j. For example, if we want to infer the author gender of cluster C_j, we divide the blog postings into groups $G_1, \ldots, G_k$ by gender. Next, we extract the writeprints, denoted by $WP(G_x)$, from each message group G_x, by employing the method described in Section 4.3.1.

6.2.3 Identifying Author Characteristics

The last step infers the characteristic of the author of anonymous messages C_j by comparing the stylometric feature items of each message ω in C_j with the writeprint $WP(G_x)$ of every group G_x. The similarity between ω and $WP(G_x)$ is computed using Equation 6.1. Message ω is labeled with class x if $WP(G_x)$ has the highest $Score(\omega \approx WP(G_x))$. All anonymous messages C_j are characterized to label x that has the major class.

$$Score(\omega \approx WP(G_x)) = \frac{\sum_{x=1}^{p} support(MP_x|G_x)}{|WP(G_x)|} \tag{6.1}$$

where $MP = \{MP_1, \ldots, MP_p\}$ is a set of matched patterns between $WP(G_x)$ and the anonymous message ω.

6.3 Experiments and Discussion

The main objective of our experiments is to evaluate the accuracy of authorship characterization method, AuthorCharacterizer, based on the training data collected from blog postings. We use 290 stylometric features including the 285 general features and 10 gender-specific features. The 285 features are described in Section 4.2.1 while the gender-specific features are listed in Appendix II and are described in [29].

The evaluation of AuthorCharacterizer has three steps. In the first step, we develop a small robot program to collect blog postings with authors' profiles from a blogger website, group them by gender and location, and extract the writeprint of each group. For characterizing the gender class, we collect 50 postings/messages for each gender type. Thus, if the total number of suspects is n, we collect $50 \times n \times 2$ blog postings in total. The average size of each posting is about 300 words. For characterizing the location information, we collect 737 postings from Australia, 800 postings from Canada, and 775 postings from the United Kingdom. In the second step, we cluster the collected postings by stylometric features, and use 2/3 of the messages for training and 1/3 for testing. In the third step, we extract the writeprints from the training messages and characterize the testing messages. A characterization of an anonymous message is correct if AuthorCharacterizer can correctly identify the characteristic of the message.

Table 6.1 shows detailed experimental results for location identification. The actual accuracy is the percentage of records that are correctly characterized in a class. The weighted accuracy is normalized by the actual number of records having the class over the total number of records. The sum is the sum of the weighted accuracy.

Table 6.1: Experimental result for location identification

No. of Authors	Region/Location	Accuracy (%)	W. Accuracy (%)	Sum (%)
4	AU	62.31	20.26	60.44
	CA	51.28	17.95	
	UK	65.39	22.23	
8	AU	46.99	15.04	50.18
	CA	62.00	21.7	
	UK	39.52	13.44	
12	AU	40.81	13.05	43.06
	CA	50.9	17.82	
	UK	35.95	12.22	
16	AU	39.98	12.79	43.21
	CA	49.16	17.21	
	UK	38.6	13.21	
20	AU	40.39	12.92	39.13
	CA	38.13	13.35	
	UK	37.83	12.86	

The accuracy scores of identifying the gender and location are depicted in Figure 6.2 and Figure 6.3, respectively. The accuracy stays almost flat at around 60% for gender, and decreases from 60.44% to 39.13% as the number of authors increases for location. One apparent reason is the least number of classes in case of gender characterization. The results suggest that the proposed frequent-pattern-based approach best fits to two class classification problem. Another possible reason is the use of 11 gender-preferential features in addition to the general stylometric features in gender identification.

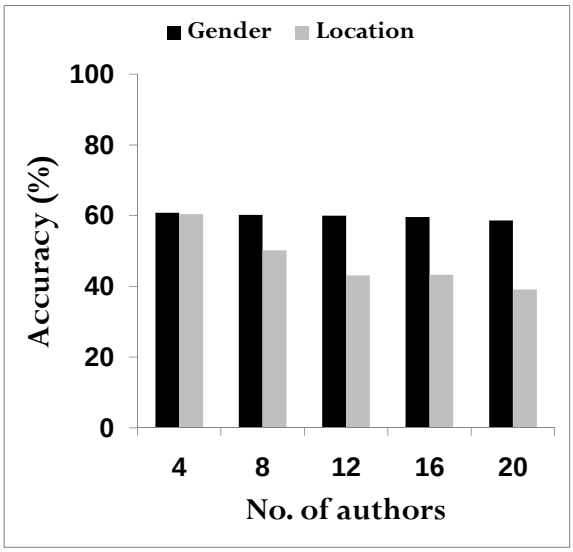


Figure 6.2: Gender identification: Accuracy vs. No. of authors

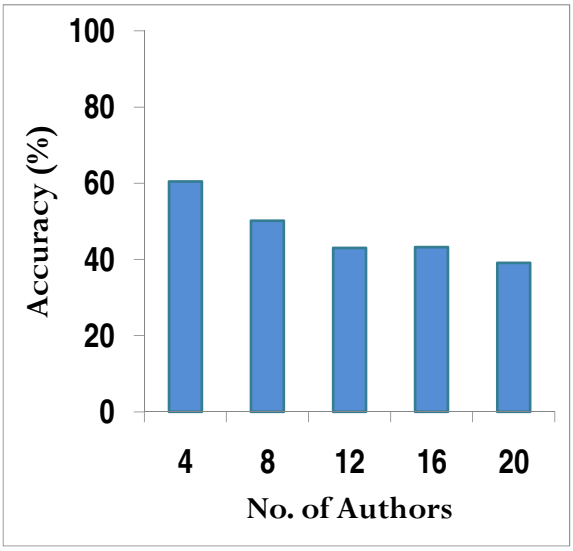


Figure 6.3: Location identification: Accuracy vs. No. of authors

6.4 Summary

In this chapter, we have developed a technique for addressing the worst case scenario of characterization problem; meaning that even the training data from the sample population is not sufficient. We evaluate the proposed method on blog dataset for two class dimensions: gender and location. In all experimental sets our method has identified the class labels correctly. Moreover, our notion of writeprint, presented in the form of frequent patterns, is suitable for forensic purposes. Experimental results on real-life data suggest that our proposed approach, together with the concept of frequent-pattern-based writeprint, is effective for identifying the author of online messages and for characterizing an unknown author.

Chapter 7

Authorship Verification

In the previous chapters, we propose methods to address two authorship problems, i.e., authorship identification and authorship characterization. In this chapter, we discuss the third authorship problem, called authorship verification. The proposed approach is applicable to different types of online messages, but in the current study we focus on e-mail messages.

The problem of authorship verification is to confirm whether or not a given suspect is the true author of a disputed textual document. Some researchers define authorship verification as a similarity detection problem, especially in cases of plagiarism. In such situation, an investigator needs to decide whether or not the two given objects are produced by the same entity. The object in question can be a piece of code, a textual document, or an online message. More importantly, the conclusion drawn needs not only to be precise but to be supported by strong presentable evidence as well.

The problems of authorship attribution and characterization, discussed in previous

chapters, are relatively well-defined, but authorship verification is not. Sometimes, it is considered as a one-class text classification problem while at another time as a two-class classification problem. Some studies address the problem by determining the dissimilarities between the writing styles of the suspect and a pseudo-suspect. Next, the metrics employed for measuring verification result vary from study to study. The measures include ROC curves [107], precision, recall [33], p-test, and t-test [121].

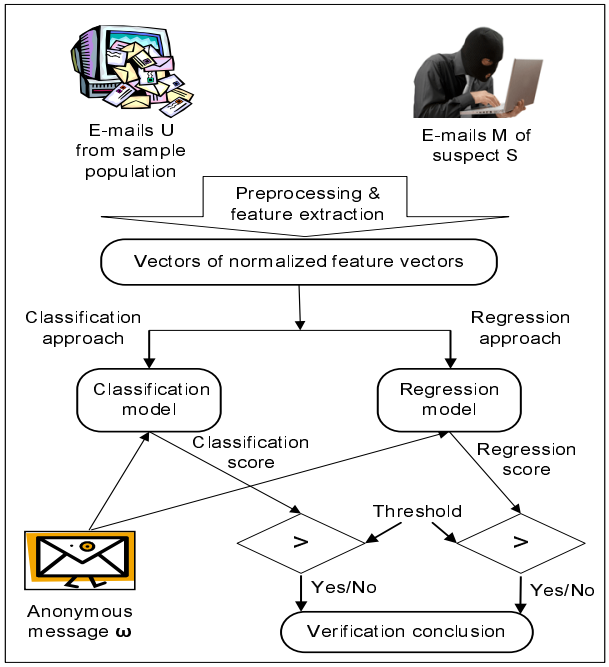


Figure 7.1: Overview of author verification approach

In this chapter, we formally define the problem of authorship verification and propose an authorship verification framework for e-mails. Our method is primarily based on the speaker recognition evaluation (SRE) framework developed by the National Institute of Standards and Technology (NIST) [72], which has proven very successful in the

speech processing community. The SRE framework evaluates the performance of detection systems in terms of minDCF, false positive and false negative alarms represented by employing a detection error trade-off (DET) curve, a deviant of the receiver operating characteristic (ROC) curve (see details in Section 7.1).

The overview of the proposed approach is shown in Figure 7.1. For two e-mail datasets, one is collected from a very large sample population denoted by U, and the other is confiscated from a potential suspect S. After the necessary preprocessing steps (cleaning, tokenization, stemming, and normalization), each e-mail is converted into a vector of stylistics or stylometric features (discussed in Section 4.2.1). We apply classification and regression techniques on both datasets. In each thread of techniques the datasets are further divided into two subsets, the training and the testing sets. Two different models, one each for suspect S, called hypothesized author and the alternate hypothesis, are trained and validated.

The given anonymous e-mail is evaluated using the two models in both regression and classification threads. Unlike the usual classification where the decision is made solely on the basis of matching probability, here the decision to verify the author is based on the threshold defined for the hypothesis testing. The threshold is calculated by varying the relative number of false positives and false negatives, depending upon the nature of the perceived application of the system. The accuracy of the system is judged in terms of EER, represented by the DET curve, and the minDCF, as using only EER can be misleading [64].

Our experimental result on a real-life data, shows that the proposed verification method has the following main contributions.

1. *Adopting NIST Speaker Recognition Framework:* We are the first to have successfully adopted the NIST's SRE framework for addressing the issue of authorship verification of textual content including e-mail dataset. We use different classification and regression methods and were able to achieve an equal error rate of 17 percent and *minDCF* equal to 0.0671 with the SVM-RBF (support vector machine-radial basis function).

2. *Employing regression for binary classification:* Regression functions, normally used for predicting numeric attributes (class labels), are employed for taking binary decision about whether or not a suspect is the author of a disputed anonymous document. It is evident from the experimental results that SVM with RBF kernel produced the best verification accuracy with the lowest *minDCF* value as compared to the classifiers used.

3. *Proposing new error detection measures for authorship verification:* To measure the performance of most detection tasks, traditionally a ROC curve is used, where false alarms are plotted against the correct detection rate. In this approach it is hard to determine the relative ratio of both types of errors, which is crucial in criminal investigation. The DET curve employed in this study can better analyze the exact contribution of both the false positive and false negative values. The use of *EER* is augmented with *minDCF* in gauging the framework accuracy.

The rest of the chapter is organized as follows: Section 7.1 defines the problem statement and different evaluation metrics. Section 7.2 presents our proposed method. Section 7.3 shows the experimental results on a real-life e-mail dataset. Section 7.4 concludes the chapter with suggestions for future work.

7.1 Problem Statement

Given a set of sample e-mails of a potential suspect S and an e-mail dataset U collected from a very large population of authors, the task of an investigator is to verify whether or not the disputed anonymous e-mail ω is written by the suspect S. Mathematically, the task of author verification can be termed as a basic hypothesis test between

H_0: ω is written by the hypothesized author S

and

H_1: ω is not written by the hypothesized author S.

The optimum test to decide between these two hypotheses is a likelihood ratio test given by

$$\frac{p(\mu|H_0)}{p(\mu|H_1)} \geq \theta \tag{7.1}$$

accept H_0, otherwise reject H_0 (accept H_1) where $p(\mu|H_i),\ i = 0, 1$ is the probability density function for the hypothesis H_i evaluated for the observed e-mail ω and θ is the decision threshold for accepting or rejecting H_0. The basic goal is to find techniques for calculating the two likelihood functions $p(\mu|H_0)$ and $p(\mu|H_1)$.

The author-specific model H_0 is well-defined and is built using e-mails written by the hypothesized author while the model H_1 is not well-defined as (potentially) it must represent the entire space of the possible alternatives to the hypothesized author.

In order to define H_1 model, we borrow the techniques used in the speaker verification literature. Two main approaches have been in use for the alternative hypothesis modeling in the speaker recognition research. The first approach is to use a set of *other-author* models to cover the space of the alternative hypothesis. This set of authors is called the cohort or the background authors. Given a set of N background author models $\lambda_1, \lambda_2, \cdots \lambda_N$, the alternative hypothesis model is represented by

$$p(\mu|H_1) = f(p(\mu|\lambda_1), p(\mu|\lambda_2), \cdots, p(\mu|\lambda_N)) \tag{7.2}$$

where $f(.)$ is some function, such as average or maximum, of the likelihood values from the background author set. The selection, size and combination of the background authors can be the subject of further research.

Another approach is the alternative hypothesis modeling in which a model is developed on sample documents are collected from a very large number of individuals. The model developed in this way is called the universal background model (UBM) in the speech processing community. We adopted the same approach for online textual documents. Given a collection of e-mail samples from a very large number of authors, a single model is trained to represent the alternative hypothesis. The main advantage of this approach is that a single author-independent model can be trained once for a particular task and then used for all hypothesized authors in that task.

Two types of errors can occur in the author verification system namely false rejection (rejecting a valid author) and false acceptance (accepting an invalid author). The probability of these errors called false rejection probability P_{fr} and false alarm probability P_{fa}. Both types of error depend on the value of user defined threshold θ. It is, therefore, possible to represent the performance of the system by plotting P_{fa} versus P_{fr}, the curve generally known as DET curve in the speech processing community.

In order to judge the performance of the author verification systems, different performance measures can be used. We borrow the two main measures namely Equal Error Rate (EER) and Detection Cost Function (DCF) from the speech processing community. The EER corresponds to the point on the DET curve where $P_{fa} = P_{fr}$. Since using only EER can be misleading [64], we use the DCF in conjunction with EER to judge the performance of author verification system.

The DCF, defined in Equation 7.3, is the weighted sum of miss and false alarm probabilities [72]. The *minDCF* means the minimum value of Equation 7.3. The DET curve is used to represent the number of false positives versus false negatives. The point on the DET curve where the number of both the false alarms become equal is called EER. The closer the DET curve to the origin, the minimum EER is and thus the better the system is.

$$DCF = C_{fr} \times P_{fr} \times P_{target} + C_{fa} \times P_{fa} \times (1 - P_{target}) \tag{7.3}$$

The parameters of the cost function are the relative costs of detection errors, C_{fr} and C_{fa} and the *a priori* probability of the specified target author, P_{target}. In our method, we

use the parameter values as specified in the NIST's SRE framework. These values are $C_{fr} = 10$, $C_{fa} = 1$ and $P_{target} = 0.01$.

The minimum cost detection function (minDCF) is redefined as the minimum value of *'0.1×false rejection rate + 0.99×false acceptance rate'*. Since it is primarily dependent on the false acceptance rate and false rejection rate and has nothing to do specifically with the speech, it can be used for the authorship verification as well. It is in conformance with the forensic analysis and strictly penalizes the false acceptance rate as it would implicate an innocent person as the perpetrator.

7.2 Proposed Approach

In this thesis, we have addressed the authorship verification as a two-class classification problem by building two models one from e-mails of the potential suspect and the other from a very large e-mail dataset belonging to different individuals called universal background model. To train and validate the two representative models, we borrowed the techniques from the SRE framework [72]. The framework is initiated by the National Institute of Standards and Technology. The purpose of the SRE framework is not only to develop state-of-the-art frameworks for addressing the issues of speaker identification and verification but to standardize and specify a common evaluation platform for judging the performance of these systems as well.

The evaluation measures such as DCF, minDCF, and EER that are used in the SRE framework are more tailored to forensic analysis as compared to simple ROC and classification accuracies. Another reason for borrowing ideas from the speaker recognition

community is that this area has a long and rich scientific basis with more than 30 years of research, development and evaluation [72]. The objective of both authorship and speaker verification is the same, i.e., to find whether or not a particular unknown object is produced by a particular subject. The object in our case is the anonymous e-mail whereas in case of speaker verification it is the speech segment. The subject is the speaker in their case whereas it is the author in our case.

As depicted in Figure 7.1, the proposed method is a two step process: model development and model application. In the first step, the given sample data is used to develop and validate the classification model. Next, the disputed anonymous message is matched with the model to verify its true author. Prior to model development, the given sample messages are converted into features vectors. The features used in the current study are described in Section 4.2.1.

To confirm whether a given anonymous e-mail ω belongs to the hypothesized author (or suspect *S*) or not, is based on the scores produced by e-mail ω during the classification process and the threshold θ. The threshold is defined by the user and is employed for taking binary decision. As described in the following paragraphs, we use two approaches for binary classification of e-mails.

7.2.1 Verification by Classification

In this approach the e-mails in the training set corresponding to the hypothesized author, and that belonging to the sample population, are nominally labeled. During the testing phase, a score is assigned to each e-mail on the basis of the probability assigned to the

e-mail by the classifier. The scores calculated for the true author and the 'imposters' are evaluated for the false acceptance and false rejection rates through a DET plot.

We use three different classification techniques, namely Adaboost.M1 [115], Discriminative Multinomial Naive Bayes (DMNB) [103] and Bayesian Network [45] classifiers. Most of the commonly used classification techniques including the one employed in the current study are implemented in the WEKA toolkit [111].

7.2.2 Verification by Regression

Authorship verification is conceptually a classification problem but in our case we need to take a binary decision of whether or not the message under test belongs to the potential suspect. Similarly, as the decision is taken on the basis of the similarity score assigned to the e-mail under test, we employ regression functions to calculate the score. We use three different regression techniques including linear regression [111], SVM with Sequential Minimum Optimization (SMO) [84], and SVM with RBF kernel [19]. We use regression scores for the true authors and for the impostors to calculating equal error rate and minimum detection cost function.

We assign an integer to e-mails of the true author and those belong to the 'imposters'. For instance, $+10$ is assigned to the hypothesized author's e-mails and -10 to e-mails of the target population. When have applied, the regression function assigns a value generally between $+10$ and -10 to the disputed anonymous e-mail. The decision, whether or not it belongs to the hypothesized author, is based on the resultant score and the user defined threshold θ.

Setting the threshold too low will increase the false alarm probability whereas setting it too high will have high miss probability (false rejection rate). In order to decide about the optimum value of the threshold and to judge the performance of our verification system, we plot the variation of the false alarm rate with the false rejection rate. The curve is generally known as the detection error trade off curve, which is drawn on a deviate scale [72]. The closer the curve to the origin, the better the verification system is. The point on the curve where the false alarm rate equals the false rejection rate is called the equal error rate.

7.3 Experiments and Discussion

To evaluate our implementation, we performed experiments on the Enron e-mail corpus made available by MIT. First, we created a universal background model from the entire Enron e-mail corpus. This is an author-independent model and is used as the basis for taking the decision whether or not the e-mail in question belongs to the suspected author. A separate model is created for each author. For this, we use 200 e-mails per author.

The decision whether an e-mail under test belongs to the hypothesized author or not is based on the difference of similarity of the e-mail to the author-independent model and that to the hypothesized author model. Based on this similarity metric, a score is assigned to the disputed e-mail. For evaluation of our classification methods, we employed the widely used 10-fold cross-validation approach by reserving 90% for training and 10% for testing. The reason is to avoid any biaseness in during the evaluation process and to judge the classification method over the entire database.

One of the performance measure used in the SRE framework is to calculate the equal error rate [72]. The EER is calculated by taking two types of scores as input namely the true author score and the false author score, which in turn are calculated by the classification methods applied over the test dataset.

Verification by Classification. Figure 7.2, depicts the DET plot of the classification results of one author, randomly selected from our database. Usually, the closer the DET curve to the origin, the minimum the EER is and thus the better the system is. The point on the DET plot which gives the minimum cost detection function is marked as a small circle on each curve.

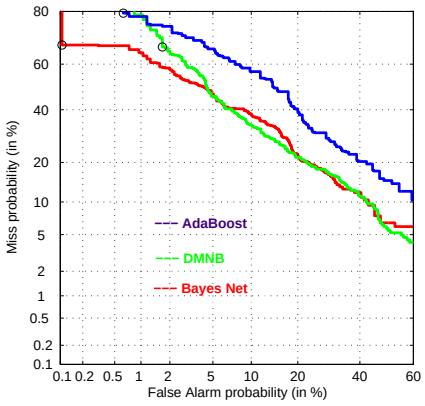


Figure 7.2: DET for author verification using classification techniques

The DET curve plotted for Bayesian Network (BayesNet) is more consistent and indicates better results both in terms of equal error rate and minimum cost detection function with less complexity. The value of *minDCF* for both DMNB and AdaBoost is comparable, however, performance of DMNB in terms of EER is closed to BayesNet. The performance gap between the two classifiers is consistent in most of the experiments.

Verification by Regression. Figure 7.3 shows the typical DET plot of one of the randomly selected author from our database, constructed by using the scores obtained from the three regression techniques as described above. The DET curve indicates that the regression approach usually produce better results in terms of *EER* and *minDCF* as compared to the classification approach. The regression approach via SVM with RBF kernel with *EER* 17.1% outperformed linear regression (with *EER* = 19.3%) and SVM-SMO (with *EER* = 22.3%). The same tendency of performance can be seen in *minDCF* values as well (see the last row of Table 7.1). DET curves for linear regression and SVM-SMO are running neck to neck starting with a highest value of false negative.

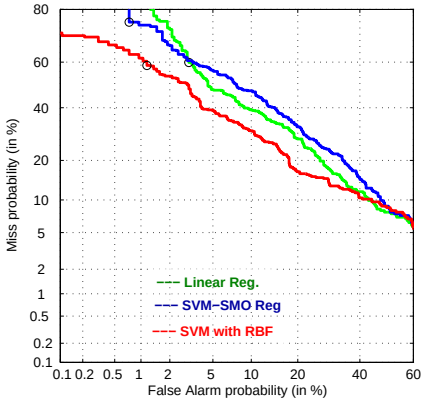


Figure 7.3: DET for author verification using regression techniques

Table 7.1: Verification scores of classification and regression methods

	Classification			***Regression***		
Verification	A.Boost	DMNB	Bayes	SVM-SMO	Lin. Reg	SVM-RBF
EER(%)	22.4	20.1	19.4	22.3	19.3	17.1
minDCF	0.0836	0.0858	0.0693	0.0921	0.0840	0.0671

The conclusion is that SVM with RBF kernel produced the best verification accuracy with the lowest minDCF value. These results suggest that regression techniques are more suitable in addressing verification problem than classifiers which perform better in attribution issues. However, the same assumption may not be always true and the result may change depending on the dataset as well as feature set used.

7.4 Summary

We have studied the problem of e-mail authorship verification and presented a solution by adopting the NIST speaker verification framework and the accuracy measuring methods. The problem has been addressed as a two-class classification problem by building two models one from e-mails of the potential suspect and the other from a very large e-mail dataset belonging to different individuals called universal background model. Experiments on a real-life dataset produces an equal error rate of 17% by employing support vector machines with RBF kernel, a regression function. The results are comparable with other state-of-the-art verification methods. Building a true 'universal' background model is not an easy task due to the non-availability of sufficient sample e-mails. The style variation of the same suspect with the changing state of mind and the context in which he writes may affect his representative model.

Chapter 8

Criminal Information Mining

In the previous chapters we have discussed about the different aspects of the authorship analysis problem, while in this chapter we propose a framework to extract criminal information from the textual content of suspicious online messages. Archives of online messages, including chat logs, e-mails, web forums, and blogs, often contain an enormous amount of forensically relevant information about potential suspects and their illegitimate activities. Such information is usually found either in the header or body of an online document.

The IP addresses, host names, sender and recipient addresses contained in the e-mail header, the user ID used in chatting, and the screen names used in web-based communication help reveal information at the user or application level. For instance, information extracted from a suspicious e-mail corpus helps us learn who the senders and recipients are, how often they communicate, how many types of communities/cliques are there in a dataset, and what are the inter- and intra-community patterns of communication. A clique

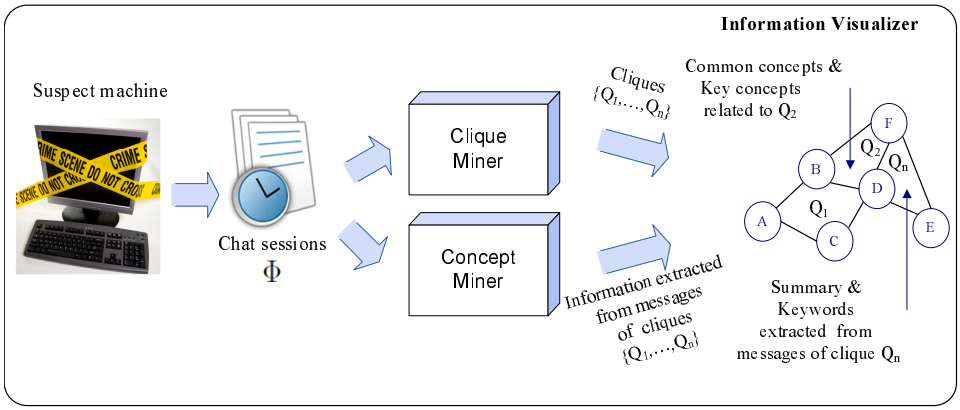


Figure 8.1: Framework overview

or a community is a group of users having an online communication link between them. Header-content or user level information is easy to extract and straightforward to use for investigation.

The focus of this study is, to analyze the *content or body* of online messages for extracting social networks and the users' topic of discussion. In this context, the problem is defined as follows: Given a suspect machine, confiscated from a crime scene, an investigator is asked to identify potential suspects who are associated with the primary suspect S and to analyze the content of online documents exchanged between suspects. The current study is focused on analyzing chat logs. The investigator is provided with a taxonomy of certain street terms, representing certain crimes, that are generally found in cybercrime-mediated textual conversation.

Though some studies on forensic analysis of online messages do exist, most of them focus on only one small aspect of the cybercrime investigation process. For instance, [65] focus on mining chat logs for collecting sociolinguistic characteristics of potential authors of anonymous chat documents. The aim of [34, 122] is to develop a classification model

for predicting the true author of an anonymous e-mail message. Alfonseca and Manandhar [8] applied *named entity recognition*, a subtask of information extraction, to extract information such as names of persons, organizations, places, or other contact information from textual documents. Minkov et al. [76] developed a technique for extracting named-entity information from informal documents such as e-mails. Chau et al. [95] proposed criminal link analysis techniques, and Xiang et al. [112] propose crime data visualization techniques for the Web- and Internet-level communication of cybercriminals. In [25], a data mining framework is developed for analyzing different kinds of crimes.

In contrast, in this study we develop a framework for extracting and reporting forensically relevant information from malicious online textual communication documents. More importantly, the entire process is automated, including retrieving documents, extracting different kinds of information and presenting the findings in an intuitive way.

The proposed framework consists of three modules including *clique miner*, *concept miner*, and *information visualizer*, as depicted in Figure 8.1. A clique, defined in this context, is a group of entities co-occurring together in the textual contents of online messages. The clique miner is designed to: first, identify the named entities appearing in the given suspicious chat logs; second, group them according to the frequency of their co-occurrence. For the former part, we use Stanford Named Entity Recognizer [1], while the latter part is accomplished by employing the concept of *frequent pattern mining* [7].

Once the cliques are extracted, the *concept miner* retrieves documents of each clique and extracts key concepts that reflect the theme of communication between members of that clique. The output of concept miner is a list of important terms (keywords), common

[1] http://nlp.stanford.edu/software/CRF-NER.shtml

concepts, key concepts, and a brief summary.

The information visualizer is used to objectively display the identified groups and the extracted information (e.g., keywords and concepts) by employing social networking concepts. In the visualized social network, depicted in Figure 8.1, the nodes represent the entities while the arcs connecting the nodes indicate the existence of a relationship between the entities. The nodes and the arcs belonging to the same clique are labeled with the letter Q_i; the subscript i indicates the clique number. The cliques are labeled with the chat summary, keywords, common concepts, and key concepts extracted from the chat sessions of a specific group.

The contributions of our study are listed below.

- *Analyzing unstructured data:* Most previous data mining methods for criminal investigation focus on structured data, e.g., criminal police records. Our data mining framework is designed for analyzing online messages including chat logs.

- *Identifying topics dynamically:* Most topic identification methods assume to have some predefined topic categories with example documents. Our approach does not need any training data and can be employed to dynamically assign topics to unknown online messages based solely on the content of the documents in question.

- *Adapting domain knowledge:* By employing the presented approach, the investigator can incorporate domain-specific terms to obtain more specific results.

The rest of the chapter is organized as follows: Section 8.1 presents the problem definition and Section 8.2 presents the proposed data mining framework. Section 8.3

evaluates the proposed approach by presenting experimental results. Section 8.4 concludes the chapter.

8.1 Problem Statement

Suppose an investigator has seized a computer from a suspect S. Let Φ be the chat log obtained from some commonly used instant messaging systems, such as Windows Live Messenger, Yahoo! Messenger, or IRC, in the computer. Typically, a chat log consists of a set of chat sessions, where each chat session contains a set of text messages exchanged between suspect S and the chat users who appear in the friend list of S. The *problem of criminal clique mining* is to discover the communities (i.e., cliques) actively involved by the suspect S in Φ, identify the relationships among the members in the cliques, and extract the concepts/topics that bring the cliques together. We divide it into two subproblems: *clique mining* and *concept analysis*.

8.1.1 Subproblem: Clique Mining

The subproblem of clique mining is to *efficiently* identify *all* the cliques from a given chat log. The following intuition of clique is formulated after an extensive discussion with the digital forensic team of a Canadian law enforcement unit. An *entity* can generally refer to the name of a person, a company, or an object identified in a chat log. To ease the discussion, we assume an entity refers to a person's name in the rest of the chapter.

A group of entities is considered to be a clique in a chat log if they chat with each other frequently, or if their names appear together frequently in some minimum number of chat sessions.

This notion of clique is more general than simply counting the number of messages sent between two chat users. An entity ε is considered to be in a clique as long as his/her name frequently appears in the chat sessions together with some group of chat users, even if ε has never chatted with the other members in the clique or even if ε is not a chat user in the log. Capturing such generalized notion of clique is important for real-life investigation because the members in a clique are not limited to be the chat users found in the log. Such generalized notion often leads to new clues for further investigation. For example, two suspected entities ε_1 and ε_2 frequently mention about the name of a third person ε_3 in the chat because ε_3 is their "boss" behind the scene. Thus, all three of them form a clique although ε_3 may not be a user found in the chat log. Nonetheless, such relaxed notion of clique may increase the chance of identifying some false positive cliques. For example, two suspects may frequently discuss about ε_3 who is a celebrity. Yet, in the context of crime investigation, an investigator would rather spend more time to filter out false positives than to miss any potential useful evidence.

A chat log Φ is a collection of chat sessions $\{\phi_1, \ldots, \phi_p\}$. Let $E(\Phi) = \{\varepsilon_1, \ldots, \varepsilon_u\}$ denote the universe of all entities identified in Φ. Let $E(\phi_i)$ denote the set of entities identified in a chat session ϕ_i, where $E(\phi_i) \subseteq E(\Phi)$. For example, $E(\phi_5) = \{\varepsilon_4, \varepsilon_5, \varepsilon_7\}$ in Table 8.1. Let $Y \subseteq E(\Phi)$ be a set of entities called *entityset*. A session ϕ_i contains an entityset Y if $Y \subseteq E(\phi_i)$. An entityset that contains k entities is called a *k-entityset*. For

Table 8.1: Vectors of entities representing chat sessions

Chat session	Identified entities
ϕ_1	$\{\varepsilon_2, \varepsilon_5, \varepsilon_7, \varepsilon_9\}$
ϕ_2	$\{\varepsilon_2, \varepsilon_5, \varepsilon_7\}$
ϕ_3	$\{\varepsilon_2, \varepsilon_5\}$
ϕ_4	$\{\varepsilon_1, \varepsilon_5, \varepsilon_7\}$
ϕ_5	$\{\varepsilon_4, \varepsilon_5, \varepsilon_7\}$
ϕ_6	$\{\varepsilon_3, \varepsilon_6, \varepsilon_8\}$
ϕ_7	$\{\varepsilon_4, \varepsilon_5, \varepsilon_8\}$
ϕ_8	$\{\varepsilon_3, \varepsilon_6, \varepsilon_8\}$
ϕ_9	$\{\varepsilon_2, \varepsilon_5, \varepsilon_8\}$
ϕ_{10}	$\{\varepsilon_1, \varepsilon_5, \varepsilon_7, \varepsilon_8, \varepsilon_9\}$

example, the entityset $Y = \{\varepsilon_3, \varepsilon_6, \varepsilon_7\}$ is a 3-entityset. The *support* of an entityset Y is the percentage of chat sessions in Φ that contain Y. An entityset Y is a clique in Φ if the support of Y is greater than or equal to some user-specified minimum support threshold.

Definition 8.1.1 (Clique)**.** Let Φ be a collection of chat sessions. Let $support(Y)$ be the percentage of sessions in Φ that contain an entityset Y, where $Y \subseteq E(\Phi)$. An entityset Y is a clique in Φ if $support(Y) \geq min\_sup$, where the minimum support threshold min_sup is a real number in an interval of [0, 1]. A clique containing k entities is called a *k-clique*. ■

Example 8.1.1. Consider Table 8.1. Suppose the user-specified threshold $min\_sup = 0.3$, which means that an entityset Y is a clique if at least 3 out of the 10 sessions contain all entities in Y. Similarly, $\{\varepsilon_4, \varepsilon_5\}$ is not a clique because it has support 2/10 = 0.2. $\{\varepsilon_2, \varepsilon_5\}$ is a 2-clique because it has support 4/10 = 0.4 and contains 2 entities. Likewise, $\{\varepsilon_5, \varepsilon_8\}$ is a 2-clique with support 3/10 = 0.3. ■

Definition 8.1.2 (Clique mining)**.** Let Φ be a collection of chat sessions. Let min_sup be a user-specified minimum support threshold. The subproblem of *clique mining* is to

efficiently identify *all* cliques in Φ with respect to *min_sup*. ■

8.1.2 Subproblem: Concept Analysis

According to the discussions with the Canadian law enforcement unit, they encountered some cases that involved thousands of chat users in the Windows Live Messenger chat log on a single machine. Consequently, there could be hundreds of cliques discovered in the chat log. The discovered cliques reflect different social aspects of the suspect, including his/her family, friendship, work, and religion. To identify the cliques related to criminal activities, the investigator has to analyze the content of the chat sessions of each clique. The subproblem of concept analysis is to extract the concepts that reflect the semantic, not just a collection of keywords, of the underlying chat conversations. To facilitate the process of concept analysis, we assume that there exists a lexical database that captures the conceptual hierarchies of a language, e.g., WordNet [48] for English.

Definition 8.1.3 (Concept analysis)**.** Let Q be a set of cliques discovered in Φ according to Definition 8.1.2. Let $\Phi(Q_i) \subseteq \Phi$ be the set of chat sessions contributing to the support of a clique $Q_i \in Q$. Note that the same chat session may contribute to multiple cliques. Let H be a lexical database of the same language used in Φ. The subproblem of *concept analysis* is to extract a set of key concepts, denoted by $KC(Q_i)$, for each discovered clique $Q_i \in Q$ by using the lexical database H. The key concepts represent the topics that bring the group of entities to form a clique. ■

8.2 Proposed Approach

Figure 8.2 depicts an overview of our proposed framework, which consists of three components including *clique miner*, *concept miner*, and *information visualizer*. *Clique miner* identifies all the cliques and their support from the given chat log. *Concept miner* analyzes the chat sessions of each identified clique and extracts the key concepts of the conversations. *Information visualizer* provides a graphical interface to allow the user to interactively browse cliques at different abstraction levels. Each module is described separately in the following paragraphs.

8.2.1 Clique Miner

The process of clique mining consists of three steps:

(1) *Dividing chat log into sessions:* A session is a sequence of messages exchanged between a group of chat users within a "logical" period of time. For instance, in the Windows Live Messenger, a session with a person P begins when the first message is sent between P and the suspect S, and ends when the suspect closes the chat log window with P. Once the chat log window is closed, re-initiating the chat is considered to be a new session with a new session ID in the log. In case of the IRC log on a public chat room, the situation is more complicated because multiple users can chat simultaneously and there are no logical break points for breaking a log into sessions. A simple solution is to break the log into sessions by some predefined unit of time, say by 15 minutes. A better solution is to look for time gap between messages and consider a new session when the time gap is larger than a short period of time, for example, 1 minute.

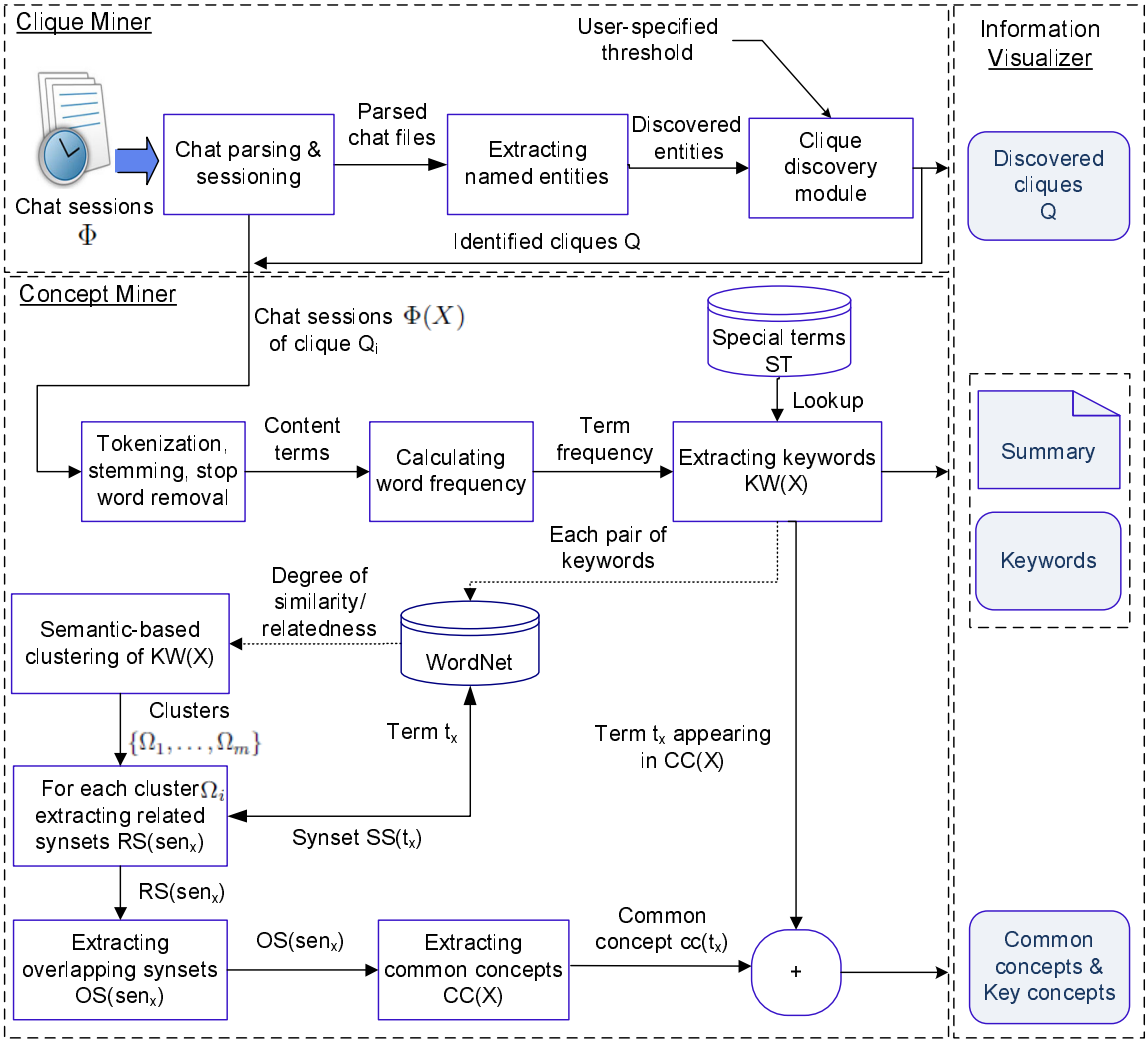


Figure 8.2: Detailed diagram of the proposed criminal information mining framework

(2) *Extracting entities:* Next, we employ the existing Named Entity Recognition (NER) tools to extract entity names from each chat session. In this study, we assume an entity is a person, but in real-life application, an entity can also be an organization, a location, a phone number, or a website [8]. NER systems use linguistic grammar-based techniques and statistical models. Hand-crafted grammar-based systems typically obtain better results, but at the cost of months of work by experienced computational linguists. Statistical NER systems typically require a large amount of manually annotated training data. In our study, we use Stanford Named Entity Recognizer[2] software called *CRF-Classifier*, which is based on the linear chain Conditional Random Field (CRF) sequence models [41]. It is trained on the widely used named entity corpora. Other NER tools can be employed if the document files contain non-English names as NER is not the focus of this study. The next step, clique mining, operates on a data table consisting of records of entities that represents entities in session, not on the actual chat log.

(3) *Mining cliques:* Recall that an entityset Y is any combination of entities identified in the chat log. An entityset is a clique if its support is equal to or greater than a given threshold. A naive approach is to enumerate all possible entitysets and identify the cliques by counting the support of each entityset in Φ. Yet, in case the number of identified entities $|E(\Phi)|$ is large, it is infeasible to enumerate all possible entitysets because there are $2^{|E(\Phi)|}$ possible combinations. We modify the Apriori algorithm [7], which is originally designed to extract frequent patterns from transaction data, to efficiently extract all cliques from Φ. We describe the modified algorithm as follows.

Recall that $E(\Phi)$ denotes the universe of all entities in Φ, and $E(\phi_i)$ denotes the set

[2]http://nlp.stanford.edu/software/CRF-NER.shtml

of entities in a session $\phi_i \in \Phi$, where $E(\phi_i) \subseteq E(\Phi)$. Our proposed Clique Miner (CM) is a level-wise iterative search algorithm that uses the k-cliques to explore the $(k+1)$-cliques. The generation of $(k+1)$-cliques from k-cliques is based on the following CM property.

Property 8.2.1 (CM property)**.** *All nonempty subsets of a clique are also cliques because* $support(Y') \geq support(Y)$ *if* $Y' \subseteq Y$. ■

By definition, an entityset Y is not a clique if $support(Y) < min\_sup$. The above property implies that adding an entity to an entityset that is not a clique will never make the entityset to become a clique. Thus, if a k-entityset Y is not an entityset, then there is no need to generate $(k+1)$-entityset $Y \cup \{\varepsilon\}$ because $Y \cup \{\varepsilon\}$ must not be a clique. The closeness among the entities in a clique Y is indicated by $|\Phi(Y)|$, which is the support of Y. CM can identify *all* cliques by efficiently pruning the entitysets that are not cliques based on the CM property.

Algorithm 8.3 summarizes our proposed Clique Mining Algorithm (CM). The algorithm identifies the k-cliques from the $(k-1)$-cliques based on the CM property. The first step is to find the set of 1-cliques, denoted by Q_1. This is achieved by scanning the chat log data table once and calculating the support count for each 1-clique. Q_1 contains all 1-cliques X with $support(C_j) \geq min\_sup$. The set of 1-cliques is then used to identify the set of candidate 2-cliques, denoted by $Candidates_2$. Then the algorithm scans the table once to count the support of each candidate X in $Candidates_2$. All candidates X that satisfy $|\Phi(X)| \geq min\_sup$ (i.e., having support greater than or equal to a threshold) are 2-cliques, denoted by Q_2. The algorithm repeats the process of generating Q_k from Q_{k-1} and stops if $Candidate_k$ is empty.

Input: Chat log Φ
Input: Minimum support threshold *min_sup*
Output: Cliques $Q = \{Q_1 \cup \cdots \cup Q_k\}$
Output: Chat sessions $\Phi(X)$, $\forall X \in Q$
1: $Q_1 \leftarrow \{\varepsilon \mid \varepsilon \in E(\Phi) \Lambda\ support(\{\varepsilon\}) \geq min\_sup\}$;
2: **for** ($k = 2$; $Q_{k-1} \neq \emptyset$; k++) **do**
3: $Candidates_k \leftarrow Q_{k-1} \bowtie Q_{k-1}$;
4: **for all** entityset $Y \in Candidates_k$ **do**
5: **if** $\exists Y' \subset Y$ such that $Y' \notin Q_{k-1}$ **then**
6: $Candidates_k \leftarrow Candidates_k - Y$;
7: **end if**
8: **end for**
9: $\Phi(X) \leftarrow \emptyset$, $\forall X \in Candidates_k$;
10: **for all** chat session $\phi \in \Phi$ **do**
11: **for all** entityset $X \in Candidates_k$ **do**
12: **if** $X \subseteq E(\phi)$ **then**
13: $\Phi(X) \leftarrow \Phi(X) \cup \phi$;
14: **end if**
15: **end for**
16: **end for**
17: $Q_k \leftarrow \{X \mid X \in Candidates_k \Lambda |\Phi(X)| \geq min\_sup\}$;
18: **end for**
19: $Q = \{Q_1 \cup \cdots \cup Q_k\}$;
20: **return** Q and $\Phi(X)$, $\forall X \in Q$;

Algorithm 3: Clique Mining Algorithm

Lines 9-17 describe the procedure of scanning the data table and keeping track of the associated document of each clique X in $Candidates_k$. Each candidate entityset X is looked up in the entities of each chat session $E(\phi)$. If a match is found, the chat session ϕ is added to the set $\Phi(X)$. If the support $|\Phi(X)|$ is greater than or equal to the user-specified minimum threshold *min_sup*, then X is added to Q_k, the set of k-cliques with k members. The algorithm terminates when no more candidates can be generated or when none of the candidate entitysets pass the *min_sup* threshold. The algorithm returns all cliques $Q = \{Q_1 \cup \cdots \cup Q_k\}$, except for the 1-cliques, with their associated chat sessions.

The following example shows how to efficiently extract all frequent patterns.

Example 8.2.1. Consider Table 8.1 with $min\_sup = 0.3$. First, identify all the entities by scanning the table once to obtain the support of every entity. The entities having support $\geq$ 0.3 are 1-cliques $Q_1 = \{\{\varepsilon_2\}, \{\varepsilon_5\}, \{\varepsilon_7\}, \{\varepsilon_8\}\}$. Then join Q_1 with itself, i.e., $Q_1 \bowtie Q_1$, to generate the candidate set $Candidates_2 = \{\{\varepsilon_2, \varepsilon_5\}, \{\varepsilon_2, \varepsilon_7\}, \{\varepsilon_2, \varepsilon_8\}, \{\varepsilon_5, \varepsilon_7\}, \{\varepsilon_5, \varepsilon_8\}, \{\varepsilon_7, \varepsilon_8\}\}$ and scan the table once to obtain the support of every entityset in $Candidates_2$. Next, identify the 2-cliques $Q_2 = \{\{\varepsilon_2, \varepsilon_5\}, \{\varepsilon_5, \varepsilon_7\}, \{\varepsilon_5, \varepsilon_8\}\}$. Similarly, perform $Q_2 \bowtie Q_2$ to generate $Candidates_3 = \{\varepsilon_5, \varepsilon_7, \varepsilon_8\}$ and determine $Q_3 = \emptyset$. Finally, the algorithm returns Q_2 and the associated chat sessions of every clique in Q_2. ■

8.2.2 Concept Miner

This phase is to analyze the chat sessions and summarize the content into some high-level topics to facilitate effective browsing in the visualization phase. The concept miner extracts the semantic from the set of associated chat sessions $\Phi(X)$ of every clique $X \in Q$ identified by Algorithm 8.3. It is important to identify the underlying semantic of the written words as many perpetrators use different obfuscation and deception techniques to covertly conduct their illegitimate activities. Understanding the semantic and contextual meaning of online messages is difficult because they are unstructured and are usually written in para language. The abbreviations, special symbols, and visual metaphors used in malicious messages convey special meaning and are meaningful in some specific context.

Specifically, the concept miner extracts three notions from $\Phi(X)$: *Keywords* are frequent words extracted from $\Phi(X)$. *Common concepts* are high-level topics shared by the chat sessions in $\Phi(X)$. *Key concepts* are the top ranked concepts by importance.

Input: Cliques Q from Algorithm 8.3
Input: Associated chat sessions $\Phi(X), \forall X \in Q$
Input: Search terms ST
Input: Keyword threshold α
Input: Maximum number of key concepts β
Output: Keywords $KW(X), \forall X \in Q$
Output: Common concepts $CC(X), \forall X \in Q$
Output: Key concepts $KC(X), \forall X \in Q$
Output: Miscellaneous information $MiscInfo(X), \forall X \in Q$

for all $X \in Q$ **do**
  $KW(X) \leftarrow \{t \mid t \in \Phi(X) \wedge t \in ST$ or t with top α $tf\_idf(t)\}$;
  Group the terms in $KW(X)$ into clusters $\{\Omega_1, \ldots, \Omega_m\}$;
  $CC(X) \leftarrow \emptyset$;
  for all cluster $\Omega_i \in \{\Omega_1, \ldots, \Omega_m\}$ **do**
    for all term $t_x \in \Omega_i$ **do**
      $SS(t_x) \leftarrow$ synsets of t_x from WordNet;
      for all sense $sen_x \in SS(t_x)$ **do**
        $RS(sen_x) \leftarrow$ related synsets of sen_x from WordNet;
        $OS(sen_x) \leftarrow RS(sen_x) \cap RS(sen_y), \forall sen_y \in SS(t_y)$, where $\forall t_y \in \Omega_i, t_x \neq t_y$;
      end for
      $CC(X) \leftarrow CC(X) \cup OS(BestSen)$, where $BestSen \in SS(t_x)$ is the sense having the largest $|OS(BestSen)|$;
    end for
  end for
  for all common concept $cc \in CC(X)$ **do**
    $Score(cc) \leftarrow 0$;
    for all term $t \in KW(X)$ **do**
      if $t \in cc$ **then**
        $Score(cc) \leftarrow Score(cc) + tf\_idf(t)$;
      end if
    end for
    $Score(cc) \leftarrow Score(cc)/|cc|$;
  end for
  $KC(X) \leftarrow \{cc \mid cc \in CC(X)$ with top β $Score(cc)\}$;
  $MiscInfo(X) \leftarrow$ various information identified in $\Phi(X)$;
end for
return $KW(X)$, $CC(X)$, $KC(X)$, and $MiscInfo(X), \forall X \in Q$;

Algorithm 4: Concept Mining Algorithm

Algorithm 8.4 provides an overview of the Concept Mining Algorithm. For every clique $X \in Q$, we extract the keywords from $\Phi(X)$, group the keywords into clusters $\{\Omega_1, \cdots, \Omega_m\}$ by semantics, extract the common concepts *CC* among the keywords within each cluster Ω_i, and finally identify the most important ones, which are the key concepts. We elaborate these five steps as follows.

Table 8.2: *Synsets* and *direct hypernyms* of selected terms retrieved from WordNet

Term	**Synsets => Direct Hypernyms**
snow	1. snow, snowfall – (precipitation falling from clouds in the form of ice crystals) => precipitation, downfall – (the falling to earth of any form of water (rain or snow or hail or sleet or mist)) 2. snow – (a layer of snowflakes (white crystals of frozen water) covering the ground) => layer – (a relatively thin sheetlike expanse or region lying over or under another) 3. Snow, C. P. Snow, Charles Percy Snow, Baron Snow of Leicester – (English writer of novels about moral dilemmas) => writer, author – (writes (books or stories or articles or the like) professionally (for pay)) 4. snow, coke, blow, nose candy, C – (street names for cocaine) => cocaine, cocain – (a narcotic (alkaloid) extracted from coca leaves; used as a surface anesthetic or taken for pleasure)
coke	1. coke – (carbon fuel produced by distillation of coal) => fuel – (a substance that can be consumed to produce energy; "more fuel is needed during the winter months") 2. coke, Coca Cola – (Coca Cola is a trademarked cola) => cola, dope – (carbonated drink flavored with extract from kola nuts ('dope' is a southernism in the United States)) 3. coke, blow, nose candy, snow, C – (street names for cocaine) => cocaine, cocain – (a narcotic (alkaloid) extracted from coca leaves; used as a surface anesthetic or taken for pleasure)
nose candy	1. coke, blow, nose candy, snow, C – (street names for cocaine) => cocaine, cocain – (a narcotic (alkaloid) extracted from coca leaves; used as a surface anesthetic or taken for pleasure)
cocaine	1. cocaine, cocain – (a narcotic (alkaloid) extracted from coca leaves; used as a surface anesthetic or taken for pleasure => hard drug – (a narcotic that is considered relatively strong and likely to cause addiction)

We first describe some standard text mining preprocessing procedures for applying to the input chat log Φ. *Tokenization* involves breaking a sentence into a sequence of words called *terms*. *Stop word removal* is applied to remove the context-independent

words, which do not contribute to identifying the semantic of the text. Stop words include function words (e.g., 'is', 'my', 'yours', and 'below'), short words (e.g., words containing 1-3 characters), punctuation, and non-informative symbols and characters [4, 121]. *Stemming* involves converting different forms of the same word into the root word [81, 86]. For instance, the words *compute*, *computed*, *computer*, and *computing* are converted into the root word *compute*. After preprocessing, each chat session $\phi \in \Phi$ is represented as a vector of terms [93].

(1) *Extracting keywords (Line 2):* There are two kinds of keywords. A term t in $\Phi(X)$ is a keyword of X, denoted by $KW(X)$, if it appears in the list of user-specified special terms or if it occurs frequently in many chat sessions of a clique but not frequently in the chat sessions of other cliques.

- Some special terms, though may not appear frequently, are important for crime investigation. For instance, certain crime-relevant street terms such as marijuana, heroin, or opium are relevant and therefore requires more attention even though they may appear only once. To identify such special terms, we allow the investigator to specify a list of special terms, denoted by ST. In our implementation, the terms are collected from different law enforcement agencies and online sources.[3].

- A term is important in $\Phi(X)$ if it appears frequently in the chat session $\Phi(X)$ of clique $X \in Q$ but not frequently in chat session $\Phi(Y)$ of other clique $Y \in Q$, where $X \neq Y$. Intuitively, these terms can help differentiate the topic of a clique from others. To identify them, we compute the $tf - idf$ of every term as discussed

[3] `http://www.whitehousedrugpolicy.gov/streetterms/`

in Section 3.4 and add the top α of them to $KW(X)$, where α is a user-specified threshold.

The sentences containing the keywords are key sentences that can be used for summary [114].

(2) *Clustering keywords by semantics (Line 3):* The objective of this step is to group the keywords into clusters $\{\Omega_1, \ldots, \Omega_m\}$ such that the keywords in the same cluster have high similarity and the keywords in different clusters have low similarity. In the literature of natural language processing, the semantic similarity is called the paradigmatic similarity and relatedness is known as syntagmatic similarity [10]. Two words are paradigmatically similar if they can be substituted by each other in a specific context without changing too much the semantic of the sentence. For instance, the word *price* can be replaced by *cost* in the sentence "*The price of the monitor is high.*" Two words are syntagmatically similar if they often appear together, for example, the words *knife* and *cut* often appear together.

We employ agglomerative hierarchical clustering method to create the clusters [97]. The general idea is to compare every pair of terms in $KW(X)$ and iteratively merge the pairs with highest similarity. The similarity is measured by the semantic relatedness of word senses according to WordNet. Specifically, we employ the *WordNet-Similarity software* [4] to compute the paradigmatic and syntagmatic similarity. Note, it is important to first cluster the words by semantics; otherwise, it will be difficult to find common concepts in the next step.

[4] `http://search.cpan.org/dist/WordNet-Similarity/`

(3) *Extracting common concepts (Lines 4-14):* Next, we want to identify some common concepts that cover the semantic of the keywords of each cluster $\Omega_i \in \{\Omega_1, \ldots, \Omega_m\}$ by making use of the WordNet. In WordNet, every term t is associated with a set of *senses* called *synset*. Each sense contains a set of terms that represents the interpretation of the term t in a specific context. Consider Table 8.2 for example. The term *coke* has three senses (synsets). In the context of drug trafficking, *coke* means *cocaine*, but it means *drink* or *carbon fuel* in different contexts. Below, we describe how to select the most suitable sense of each term based on the context described by other terms in the same cluster.

We perform the following operations for every keyword t_x in each cluster Ω_i. First, we obtain the related synsets denoted by $RS(sen_x)$, including the synonyms, direct hypernyms, and entailments for every sense sen_x of t_x. Second, we identify the overlapping related synsets of sen_x and of every other term t_y in the same cluster Ω_i. The overlapping synsets are denoted by $OS(sen_x)$. Finally, we select the best sense, denoted by *BestSen*, that has the largest number of overlapping synset, and add $OS(BestSen)$ to the common concepts of clique X, denoted by $CC(X)$. Table 8.2 lists the senses (synsets) of some terms followed by a direct hypernym of the sense. Suppose we find the keywords *coke* and *snow* in some chat sessions of a clique. By intersecting their related synsets including the direct hypernyms, we can identify a common concept *{coke, blow, nose candy, snow, C}*, which has a direct hypernym *{hard drug}*. Without considering the terms *coke* and *snow* in the correct context, the terms will probably be misinterpreted.[5]

[5]We use the *Java API for WordNet Searching (JAWS)* `http://lyle.smu.edu/~tspell/jaws/` to retrieve the synsets from WordNet.

(4) *Identifying key concepts (Lines 15-24):* According to the evaluation, the semantic of the chat sessions associated with a clique is well-captured by the common concepts extracted. However, in real application, there are too many of them. It is impractical to display *all* the common concepts in the interactive user interface for browsing the cliques. Thus, we rank the common concepts, and display the top β of them, where β is a user-specified threshold. Intuitively, a common concept is a key concept in X if its senses contain a keyword found in the clique X. The importance of a term is computed by the tf_idf value. The importance of a common concept is the sum of the tf_idf values of the matched terms normalized by the number of terms in the common concept.

(5) *Extracting miscellaneous information (Line 25):* This step extracts some relevant information, such as phone number, addresses, e-mails, website URLs, from the chat sessions of every clique. This task can be easily achieved by matching with some regular expressions.

8.2.3 Information Visualizer

The objective of the information visualizer is to provide an interactive user interface to browse the discovered cliques and the relevant information. In general, a clique can be displayed as a graph, in which the nodes represent the entities, the edges represent the relationship, and the lengths of edges represent the closeness between the entities. Yet, the visualization task in this study is challenging when the number of discovered cliques is large. Recall that Property 8.2.1 states that every subset of a clique is also a clique, so the discovered cliques in fact represent multiple layers of relationships. Each clique

has its own closeness, keywords, common concepts, key concepts, and other relevant information. We have designed an intuitive interface by integrating a data visualization tool, called prefuse [54], which allows the user to drill-down and roll-up on a clique. Prefuse is a collection of software tools, written in Java, and is used for creating interactive data visualization solutions. See the next section for more details.

8.3 Experiments and Discussion

We have four objectives in this section. (1) To verify if the cliques, extracted by the clique miner, represent a meaningful group of individuals in real-world and to measure the effect of minimum support threshold on the number of cliques. (2) To evaluate whether or not the concept miner can precisely identify the important keywords, common concepts, and key concepts from the chat conversation of each extracted clique. (3) To quantitatively measure the efficiency of the developed framework in terms of the total execution time versus user-defined minimum support threshold. (4) To measure the scalability of the presented framework by plotting the execution time viz-a-viz data size.

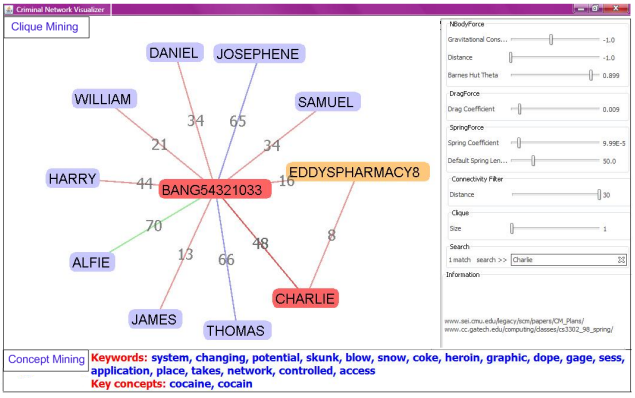


Figure 8.3: A sample screen shot of the presented framework

Finding a real-life dataset for evaluating the proposed approach is not trivial due to privacy issues. Most law enforcement agencies and private organizations, though have access to criminal data, but they can not make it public due to legal constraints. For instance, the chat corpus, collected by Perverted Justice [6], is a rich source of predominantly cybercrime related data and is available online but it cannot be used for analysis without the consent of the people concerned. These chat logs mostly contain cyber predatory and cyber bullying conversation between predators and the pseudo-victims. Therefore, our research team objectively creates MSN chat logs in which one of the team member pretends to be the primary suspect chatting with different users. In the given chat conversation, one of our member behaves as a pseudo-drug dealer by using street names of some drugs in his conversation with the primary suspect.

In the first set of experiments, the clique miner takes the given chat log and displays the identified cliques. Figure 8.3, representing the screen shot of the framework, displays the graphical view of the discovered cliques. By using the *GUI* of the developed framework, the user can identify a clique by moving the mouse on the figure. The group of entities representing one clique are highlighted together. In the figure, ten cliques, each containing 2-3 entities are shown. The central node in each clique denotes the primary suspect and the peripheral nodes represent the entities associated with the suspect. The arcs connecting the entities indicate the existence of relationship between the entities. The clique containing entities *BANG*54321033, *EDDYSPHARMACY*8, and *CHARLIE* is interesting as the chat conversation of its member contains drug-related terms, e.g., grass, pot dope, skunk, and snow. We have manually compared the extracted entities and

[6] http://perverted-justice.com/

the discovered cliques with the textual content of the given chat sessions. We found that more than 80% of the cliques are correctly identified with a few false positive cases. This can be further improved by using sophisticated tools for named entity recognition.

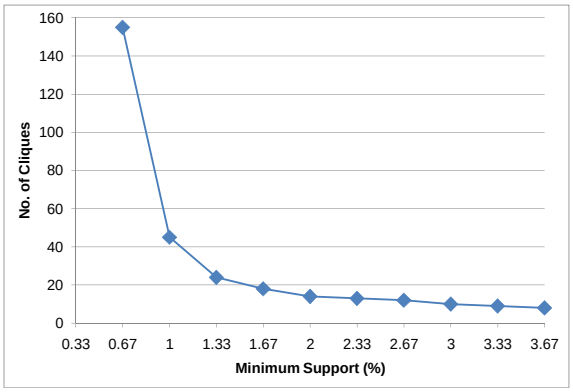


Figure 8.4: Effect of minimum support on number of cliques

In the second set of experiments, we incrementally change the user-defined minimum support threshold to check its effect on the number of cliques, as shown in Figure 8.4. The number of cliques extracted from the given chat log spans from 155 to 8 for minimum support ranging from 0.33% to 3.33%. The number of cliques is inversely proportional to the minimum support, i.e., increasing the support will cause a decrease in the number of cliques. The number of cliques sharply drops by changing the support threshold from 0.33% to 0.66% for the chat log in question. The curve becomes almost flat when the support count is increased to 1.33%. There is always a trade off between the two parameters and can be adjusted according to the specific requirements of an investigation.

The third set of experiments is performed to evaluate the concept analysis functionality of the presented framework. The concept miner retrieves the chat log of each clique,

discovered in the clique mining step, and extract the keywords, common concepts, and key concepts from each chat collection separately. Figure 8.3 visualizes the extracted cliques and the concept analysis results associated with each clique. The drill-down and roll-up capability of the framework, allows the user to browse the cliques and the summary of their conversation.

We found the concept analysis summary of the chat log belonging to the clique comprising BANG54321033, EDDYSPHARMACY8, and CHARLIE interesting. The extracted keywords including blow, snow, coke, dope, and gage, which are the street terms used to represent cocaine, a narcotic. The concept miner also identifies the words including system, changing, and potential as keywords, which happened due to the high frequency of these words. The words 'cocaine' and 'cocain', identifies as the key concepts, represent the topic of chat conversation of the aforementioned clique. The other extracted information such as the message summary and the common concepts are not shown in the figure for simplicity. By comparing the extracted keywords and the related key concepts with the WordNet conceptual hierarchy (shown in Table 8.2), suggests that the concept miner can correctly identify the topic of online messages.

The slide bars, denoted by NBodyForce in Figure 8.3, are used by the user for setting the parameters. The user needs to specify the minimum support threshold and the size of the chat dataset.

The fourth set of experiments is employed to measure the runtime efficiency of our framework. For this, we have used MSN chat logs with an initial size of 2.59MB,

voluntarily contributed by our team members. The value of *total execution time* (measured in seconds) is plotted against the minimum support, as shown in Figure 8.5. The value of minimum support ranges from 0.33% to 3.33%. The total execution time is maximum, i.e., 53 seconds for minimum support 0.66% and decreases as the minimum support increases. The execution time drops sharply from 0.66% to 1.0% and remains flat afterwards.

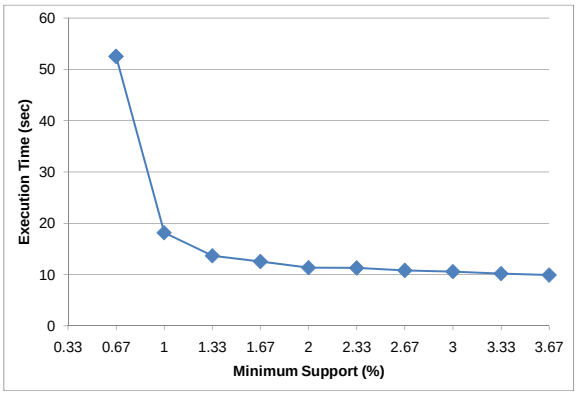


Figure 8.5: Efficiency [Execution time vs. Minimum support]

Generally, a software tool is considered scalable provided its execution time increases linearly as the size of input data increases. However, if the execution time grows exponentially with the increase in the data size, then the tool is not scalable. To measure the scalability, we incrementally change the size of the dataset (measured in terms of the total chat sessions) while keeping the minimum support constant at 0.67% in clique miner. Initially, we use a dataset of 1000 sessions, which is incremented by an equal interval size of 2000 sessions up to a maximum size of 10000 sessions. Depicted in Figure 8.6, we measure the execution time of each component of our framework separately. Finally, we add up all the individual scores together to obtain the total execution time of

the entire framework code. From this graph, we can clearly see a linear increase in the execution time of each component as well as the sum total of all the components. The figure indicates that the proposed framework is scalable.

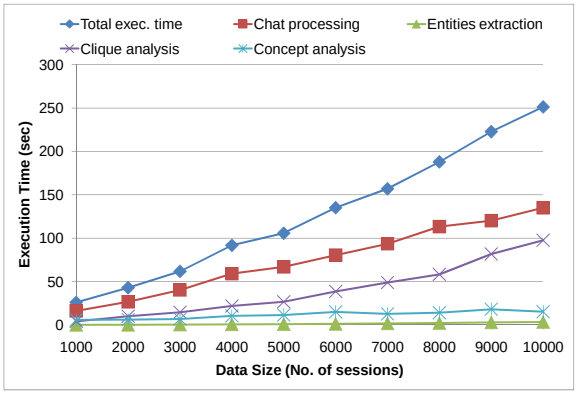


Figure 8.6: Scalability [Execution time vs. Data size]

8.4 Summary

In this chapter, we have developed a criminal information mining framework for extracting forensically relevant information from suspicious online messages. The framework is designed to take online messages as input and provides a set of cliques and the topic of discussion of the chat conversation of each clique as output. The experimental result on a given chat log suggests that the proposed framework can precisely identify the pertinent cliques and the perceived meaning of the messages exchanged between members of each clique. The framework meets the standard requirements of efficiency and scalability. The accuracy of the framework can be improved by developing a precise and efficient knowledge-base of the commonly used cybercrime terms. Moreover, the result can be improved by employing sophisticated techniques in the preprocessing step and by using

a dataset that is predominantly malicious.

Moreover, the current version of WordNet contains limited number of cybercrime-related words and therefore it needs to be extended to include more terms. Similarly, to extend the proposed framework to support languages other than English, the need is to develop a WordNet-like lexical database, e.g., EuroWordNet.

Chapter 9

Conclusion and Future Work

This chapter concludes the thesis. First, we give a summary of our presented framework and main contributions of this thesis followed by some pointers for future research.

9.1 Thesis Summary

We have developed and implemented a forensic analysis framework to help an investigator and expert witness collect empirical evidence by automatically analyzing large archives of suspicious online messages. The analysis can be performed on the header content as well as on the textual body of a message. To perform header analysis, we have implemented some state-of-the-art techniques in our framework; however, our study has focused on the message body. The message body usually contains two types of content: all-purpose content-independent words and content-specific or content-dependent words. Content-independent words, called authorial attributes, are collected from previously written messages of suspects to address the problem of anonymity. The content-specific part is used

to extract knowledge relevant to a cybercrime investigation. Therefore, the presented framework consists of two main modules: *authorship analysis* and *criminal information mining*.

In the authorship analysis module, we introduce a novel approach of authorship analysis and formulate a new notion of writeprint based on the concept of frequent pattern mining. Unlike the physical fingerprint, we do not claim that the writeprint can uniquely distinguish every individual in the world, but our experimental results strongly suggest that the writeprint defined in this study is accurate enough to uniquely identify the writing style of one individual among a limited number of suspects. Our notion of writeprint has two special properties that make it different from the traditional notion of writeprint in the literature [4, 5].

First, the *combination* of feature items forming the writeprint of a suspect is dynamically generated based solely on the embedded patterns in his/her messages. This flexibility allows us to succinctly model the writeprint of different suspects by using different combinations of feature items. In contrast, the traditional notion of writeprint considers one feature at a time without considering *all* combinations.

Second, every frequent stylometric pattern in our notion of writeprint captures a piece of writing pattern that can be found *only* in one suspect's messages, but not in other suspects' messages. A cybercrime investigator could precisely point out such matched patterns in the anonymous message to support the conclusion of authorship disputes. In contrast, the traditional classification method, e.g., decision tree, attempts to use the *same* set of features to capture the writeprint of different suspects. It is quite possible that

the classifier would capture some common writing patterns and the investigator could unintentionally use the common patterns to draw a wrong conclusion on authorship. Our notion of writeprint avoids such problem and, therefore, provides more convincing and reliable evidence.

Our method produces acceptable results in resolving the three types of authorial disputes, i.e., identification, characterization, and verification. The proposed method is effective even if there exists only a few training samples or even no training samples. The proposed data mining approach can tackle the problem of stylistic variation by capturing the sub-stylistic features of a suspect. Similarly, our experiments suggest that text messages can be divided into different groups based on writing styles by applying stylometry-based clustering.

Is the accuracy demonstrated in our experiments good enough for criminal investigations? According to our discussion with a law enforcement unit, having 70%-90% identification accuracy is acceptable in an investigation, especially in the early phase when a crime investigator often has very few clues to begin with. Yet, we emphasize that our proposed methods cannot (and should not) substitute for the role of an expert witness in a court of law. The methods can speed up the analysis process and can identify some less obvious combination of stylometric patterns, but an expert witness still has to apply his/her expert knowledge to verify the consistency of the extracted results with other available evidence.

By employing the *criminal information mining* component of our approach, an investigator would be able to extract forensically relevant information from large archives of

suspicious online messages. The experimental results on real-life dataset suggest that the implemented techniques can be used to identify suspicious entities and their hidden relationships from the messages in question. Further, the developed method is able to identify the contextual meaning of the written words, and thus is effective in retrieving messages containing malicious material. Our method can be applied for query expansion in cases where the user has a limited number of search words or her knowledge is limited in the domain of cybercrime investigation. To evaluate whether or not the implemented approach meets the standard requirements of efficiency and scalability, we measured these characteristics in terms of the total execution time. Finally, we present the extracted knowledge in a more intuitive way to facilitate the decision-making process.

We would like to share our technical expertise acquired through our team work with the local law enforcement agencies. Cybercrime investigation is complex, often a combination of technical, legal, and resource issues. To develop an effective multidisciplinary approach, it is important to educate investigators about the latest data mining technology and the tools available for crime investigation. When investigators encounter problems in criminal information mining, as presented in this study, their initial response is often to solve the problem manually or to conduct a simple search using general-purpose search engines. In fact, alternative techniques, such as the data mining solution presented in this study, are available to help significantly reduce the investigation time.

9.2 Future Work

There is still a long way to go to develop a comprehensive, reliable forensic analysis approach before it can be widely accepted in courts of law. The small size, unstructured layout, and informal contents of online messages make the analysis process more challenging.

Future research in authorship analysis can be focused on the following three areas.

(1) Our current version of AuthorMiner2 relies on an investigator to divide the messages into groups such that sub-writeprints can be derived. As a result, the identification result varies depending on the subjective grouping. One possible improvement is to devise a clustering method to group the training messages by sub-writeprints. (2) Our current approach of AuthorCharacterizer utilizes blog postings to infer characteristics of an e-mail author. Though our approach demonstrates some initial success, some stylometric features of e-mails are not applicable to blog postings. To further improve the characterization accuracy on e-mails, one research direction is to collect a large volume of sample e-mails from authors with different backgrounds, extract the writeprints, and then use the writeprints to infer the characteristics of potential suspects based on their e-mails.

(3) Our study shows that content-specific keywords can play an important role in style mining when used in specific contexts like cybercrime investigation. Therefore, it is imperative to develop a sound technique for identifying effective and significant keywords. Feature optimization for selecting the most appropriate attributes among the available approximately 1000 writing style features is another potential research direction.

Addressing language multiplicity is another research direction in analyzing online

messages. The Internet has become a common venue for cybercriminals coming from different regions and ethnic groups, speaking different languages, and following different norms and traditions. Therefore, it is very difficult to understand the underlying meaning of conversation between the perpetrators. Similarly, the ever increasing number of new obfuscation techniques used by perpetrators for hiding their suspicious online communications makes analysis of electronic discourse more difficult. Similarly, more interesting results can be obtained by using the proposed approach on real-time online traffic containing malicious messages.

To develop an effective investigation approach, an expert witness needs to acquire an up-to-date knowledge of innovative data mining and language processing techniques. Similarly, the techniques developed for analysis of structured documents are not very effective for analyzing online messages. Therefore, there is a need to design techniques that best fit the analysis of electronic discourse, the so-called written conversation, produced in different languages.

Publications in refereed journals and conferences.

Journals

- F. Iqbal, R. Hadjidj, B. C. M. Fung, and M. Debbabi. A novel approach of mining write-prints for authorship attribution in e-mail forensics. Digital Investigation (DIIN), 5(1):42-51. Elsevier. In Proc. of the Eighth Annual DFRWS Conference, September 2008.

- R. Hadjidj, M. Debbabi, H. Lounis, F. Iqbal, A. Szporer, and D. Benredjem. Towards an Integrated Email Forensics Analysis Framework. Digital Investigation

(DIIN), 5(3):124-137, March 2009. Elsevier.

- F. Iqbal, H. Binsalleeh, B. C. M. Fung, and M. Debbabi. Mining writeprints from anonymous e-mails for forensic investigation. Digital Investigation (DIIN), 7(1):56-64, October 2010. Elsevier.

- L. A. Khan, F. Iqbal, and S. M. Baig. Speaker Verification from Partially Encrypted Compressed Speech Digital Investigation. Digital Investigation, 7(1):74-80, October 2010. Elsevier.

Conferences

- F. Iqbal, L. A. Khan, B. C. M. Fung, and M. Debbabi. E-mail authorship verification for forensic investigation. In Proc. of the 25th ACM SIGAPP Symposium on Applied Computing (SAC): Computer Forensics, pages 1591-1598, Sierre, Switzerland: ACM Press, March 2010.

Articles in process for publication in refereed journals

- F. Iqbal, B. C. M. Fung, H. BinSalleeh, and M. Debbabi. A Unified Data Mining Solution for Authorship Analysis in Anonymous Textual Communications. Information Sciences: Special Issue on Data Mining for Information Security, Elsevier. (under 2nd revision).

Bibliography

[1] Network E-mail Examiner. Web site: http://www.paraben-enterprise.com/, Retrieved on August 15, 2010. Paraben Corporation.

[2] Forensic ToolKit. Web site: http://www.accessdata.com/forensictoolkit.html, Retrieved on March 2, 2009. AccessData.

[3] Encase. Web site: http://www.guidancesoftware.com/, Retrieved on May 10, 2010. Guidance Software.

[4] A. Abbasi and H. Chen. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems*, 26(2):1–29, 2008.

[5] A. Abbasi, H. Chen, and J. Nunamaker. Stylometric identification in electronic markets: Scalability and robustness. *Journal of Management Information Systems*, 5(1):49–78, 2008.

[6] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. of ACM*

SIGMOD Conference, Seattle, WA, 1998.

[7] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, Washington, D.C., United States, 1993. ACM.

[8] E. Alfonseca and S. Manandhar. An unsupervised method for general named entity recognition and automated concept discovery. In *Proc. of International Conference on General WordNet*, 2002.

[9] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study final report. In *Proc. of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218, 1998.

[10] M.-H. Antoni-Lay, G. Francopoulo, and L. Zaysser. A generic model for reusable lexicons: The genelex project. *Literary and Linguistic Computing*, 9(1), 1994.

[11] S. Argamon, M. Koppel, and G. Avneri. Routing documents according to style. In *Proc. of the First International Workshop on Innovative Information Systems*, 1998.

[12] S. Argamon and M. Saric. Style mining of electronic messages for multiple authorship discrimination: first results. In *Proc. of the 9th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 475–480, Washington, D.C., 2003. ACM.

[13] R. H. Baayen, H. van Halteren, and F. J. Tweedie. Outside the cave of shadows: using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing*, 2:110–120, 1996.

[14] R. Barzilay, N. Elhadad, and K. R. Mckeown. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55, 2002.

[15] R. Barzilay and K. R. Mckeown. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31:297–328, 2005.

[16] J. Bengel, S. Gauch, E. Mittur, and R. Vijayaraghavan. ChatTrack: Chat Room Topic Detection Using Classification. In *Proc. of the 2nd Symposium on Intelligence and Security Informatics (in review*, pages 266–277, 2004.

[17] M. Bhattacharyya, S. Hershkop, E. Eskin, and S. J. Stolfo. MET: An experimental system for malicious email tracking. In *Proc. of the 2002 New Security Paradigms Workshop (NSPW-2002)*, Virginia Beach, VA, 2002.

[18] M. D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, Second edition, 2003.

[19] C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[20] J. F. Burrows. Word patterns and story shapes: the statistical analysis of narrative style. *Literary and Linguistic Computing*, 2:61–67, 1987.

[21] J. F. Burrows. 'an ocean where each kind...': Statistical analysis and some major determinants of literary style. *Computers and the Humanities*, 23:309–321, 1989.

[22] V. R. Carvalho and W. W. Cohen. Learning to extract signature and reply lines from email. In *Proc. of the conference on email and anti-spam*, Mountain View, CA, 2004.

[23] H. Chen, W. Chung, J. Qin, E. Reid, M. Sageman, and G. Weimann. Uncovering the dark web: A case study of jihad on the web. *J. Am. Soc. Inf. Sci. Technol.*, 59(8):1347–1359, 2008.

[24] H. Chen, W. Chung, Y. Qin, M. Chau, J. J. Xu, G. Wang, R. Zheng, and H. Atabakhsh. Crime data mining: an overview and case studies. In *Proc. of the annual national conference on digital government research*, pages 1–5. Digital Government Society of North America, 2003.

[25] H. Chen, W. Chung, J. J. Xu, G. Wang, Y. Qin, and M. Chau. Crime data mining: A general framework and some examples. *Computer*, 37(4):50–56, 2004.

[26] N. A. Chinchor. Overview of MUC-7/MET-2. In *Proc. of the Seventh Message Understanding Conference (MUC-7)*. Morgan, 1998.

[27] N. Chou, R. Ledesma, Y. Teraguchi, and J. C. Mitchell. Client-side defense against web-based identity theft. In *Proc. of the Network and Distributed System Security Symposium, NDSS 2004*, San Diego, California, USA, 2004.

[28] C. E. H. Chua and J. Wareham. Fighting internet auction fraud: An assessment and proposal. *Computer*, 37:31–37, 2004.

[29] M. Corney, O. de Vel, A. Anderson, and G. Mohay. Gender-preferential text mining of e-mail discourse. In *ACSAC'02: Proc. of the 18th Annual Computer Security Applications Conference*, pages 21–27, Washington, DC, USA, 2002. IEEE Computer Society.

[30] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines*. Cambridge University Press, UK, 2000.

[31] D. Cutting, D. Karger, J. Pedersen, and J. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proc. of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–329, 1992.

[32] D. Das and A. F. T. Martins. A survey on automatic text summarization. Web site: http://www.cs.cmu.edu/ nasmith/LS2/das-martins.07.pdf, 2007. Language Technologies Institute, Carnegie Mellon University.

[33] O. de Vel. Mining e-mail authorship. In *Proc. of ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, Boston, 2000.

[34] O. de Vel, A. Anderson, M. Corney, and G. Mohay. Mining e-mail content for author identification forensics. *SIGMOD Record*, 30(4):55–64, 2001.

[35] O. de Vel, A. Anderson, M. Corney, and G. Mohay. Multi-topic e-mail authorship attribution forensics. In *Proc. of ACM Conference on Computer Security - Workshop on Data Mining for Security Applications*, 2001.

[36] O. de Vel, M. Corney, A. Anderson, and G. Mohay. Language and gender author cohort analysis of e-mail for computer forensics. In *Proc. of Digital Forensic Research Workshop*, 2002.

[37] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[38] J. Diesner and K. M. Carley. Exploration of communication networks from the enron email corpus. In *Proc. of Workshop on Link Analysis, Counterterrorism and Security, SIAM International Conference on Data Mining*, pages 21–23. SIAM, 2005.

[39] H. Dong, S. C. Hui, and Y. He. Structural analysis of chat messages for topic detection. *Online Information Review*, 30(5):496–516, 2006.

[40] E. Elnahrawy. Log-based chat room monitoring using text categorization: A comparative study. In *Proc. of the International Association of Science and Technology for Development Conference on Information and Knowledge Sharing (IKS 2002)*, pages 381–388. St. Thomas, US Virgin Islands, USA, 2002.

[41] J. R. Finkel, T. Grenager, and C. Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370, 2005.

[42] J. Foertsch. The impact of electronic networks on scholarly communication: Avenues for research. *Discourse Processes*, 19(2):301–328, 1995.

[43] R. S. Forsyth and D. I. Holmes. Feature finding for text classification. *Literary and Linguistic Computing*, 11(4):163–174, 1996.

[44] E. Frank and S. Kramer. Ensembles of nested dichotomies for multi-class problems. In *Proc. of the 21st International conference of Machine Learning (ICML-2004*, pages 305–312. ACM Press, 2004.

[45] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian Network Classifiers. *Machine Learning*, 29:131–163, 1977.

[46] B. C. M. Fung, K. Wang, and M. Ester. Hierarchical document clustering using frequent itemsets. In *Proc. of the 3rd SIAM International Conference on Data Mining (SDM)*, pages 59–70, San Francisco, CA, May 2003.

[47] M. Gamon. Linguistic correlates of style: authorship classification with deep linguistic analysis features. In *Proc. of the 20th International Conference on Computational Linguistics*, pages 611–617, Geneva, Switzerland, 2004.

[48] A. M. George. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[49] A. Gray, P. Sallis, and S. Macdonell. Software forensics: Extending authorship analysis techniques to computer programs. In *Proc. of the 3rd Biannual Conf. Int. Assoc. of Forensic Linguists (IAFL'97*, pages 1–8, 1997.

[50] R. Hadjidj, M. Debbabi, H. Lounis, F. Iqbal, A. Szporer, and D. Benredjem. Towards an integrated email forensics analysis framework. *Digital Investigation*, 5(3-4):124–137, 2009.

[51] J. Han and J. Pei. Mining frequent patterns by pattern-growth: methodology and implications. *SIGKDD Explor. Newsl.*, 2(2):14–20, 2000.

[52] C. Hansen. *To Catch a Predator: Protecting Your Kids from Online Enemies Already in Your Home*. Tantor Media, 2007.

[53] A. Hartigan and M.A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

[54] J. Heer, S. K. Card, and J. A. Landay. prefuse: a toolkit for interactive information visualization. In *Proc. of the SIGCHI conference on Human factors in computing systems*, pages 421–430, Portland, Oregon, USA, 2005. ACM.

[55] M. Hegland. The apriori algorithm - a tutorial. *WSPC/Lecture Notes Series*, 9(7), March 2005. http://www2.ims.nus.edu.sg/preprints/2005-29.pdf.

[56] D. I. Holmes. The evolution of stylometry in humanities. *Literary and Linguistic Computing*, 13(3):111–117, 1998.

[57] J. D. Holt and S. M. Chung. Efficient mining of association rules in text databases. In *Proc. of the 8th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 234–242, Kansas City, Missouri, United States, 1999. ACM.

[58] F. Iqbal, H. Binsalleeh, B. C. M. Fung, and M. Debbabi. Mining writeprints from anonymous e-mails for forensic investigation. *Digital Investigation*, pages 1–9, 2010.

[59] F. Iqbal, H. Binsalleeh, B. C. M. Fung, and M. Debbabi. Mining writeprints from anonymous e-mails for forensic investigation. *Digital Investigation*, in press.

[60] F. Iqbal, R. Hadjidj, B. C. M. Fung, and M. Debbabi. A novel approach of mining write-prints for authorship attribution in e-mail forensics. *Digital Investigation*, 5(1):42–51, 2008.

[61] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proc. of European Conf. Machine Learning (ECML'98)*, pages 137–142. Springer Verlag, 1998.

[62] T. Kolenda, L. K. Hansen, and J. Larsen. Signal detection using ICA: Application to chat room topic spotting. In *Proc. of the Third International Conference on Independent Component Analysis and Blind Source Separation*, pages 540–545, 2001.

[63] M. Koppel, S. Argamon, and A. R. Shimoni. Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 17(4):401–412, 2002.

[64] M. Koppel, J. Schler, and S. Argamon. Computational methods in authorship attribution. *J. Am. Soc. Inf. Sci. Technol.*, 60(1):9–26, 2009.

[65] T. Kucukyilmaz, B. B. Cambazoglu, F. Can, and C. Aykanat. Chat mining: predicting user and message attributes in computer-mediated communication. *Information Processing and Management*, 44(4):1448–1466, 2008.

[66] A. Kulkarni and T. Pedersen. Name discrimination and e-mail clustering using unsupervised clustering and labelling of similar contexts. In *Proc. of the 2nd Indian International Conference on Artificial Intelligence (IICAI-05)*, pages 703–722, 2005.

[67] B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In *Proc. of the 5th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 16–22, 1999.

[68] G. R. Ledger and T. V. N. Merriam. Shakespeare, Fletcher, and the two Noble Kinsmen. *Literary and Linguistic Computing*, 9:235–248, 1994.

[69] H. Li, D. Shen, B. Zhang, Z. Chen, and Q. Yang. Adding semantics to email clustering. In *Proc. of the 6th International Conference on Data Mining (ICDM)*, pages 938–942, Washington, DC, USA, 2006. IEEE Computer Society.

[70] R. P. Lippmann. An introduction to computing with neural networks. *IEEE Acoustics Speech and Signal Processing Magazine*, 4(2):4–22, 1987.

[71] L. M. Manevitz, M. Yousef, N. Cristianini, J. Shawe-taylor, and B. Williamson. One-class svms for document classification. *Journal of Machine Learning Research*, 2:139–154, 2001.

[72] A. Martin and M. Przybocki. The nist speaker recognition evaluation series. National Institute of Standards and Technology Web site, June 2009.

[73] W. McIver and A. Elmagarmid. *Advances in Digital Government: Technology, Human Factors, and Policy*. Springer, 2002.

[74] T. C. Mendenhall. The characteristic curves of composition. *Science*, 11(11):237–249, 1887.

[75] A. Miller. *Subset Selection in Regression*. Chapman & Hall/CRC, 2002.

[76] E. Minkov, R.C. Wang, and Cohen W. W. Extracting personal names from email: Applying named entity recognition to informal text. In *HLT/EMNLP*, 2005.

[77] F. Mosteller and D. L. Wallace. *Applied Bayesian and classical inference: The case of the Federalist Papers*. Springer-Verlag, New York, Second edition, 1964.

[78] F. Mosteller and D. L. Wallace. *Inference and Disputed Authorship: The Federalist.* behavioral science:quantitative methods edition. Addison-Wesley, Massachusetts, 1964.

[79] J. Novak, P. Raghavan, and A. Tomkins. Anti-aliasing on the web. In *Proc. of the 13th international conference on World Wide Web*, pages 30–39. ACM, 2004.

[80] Ö. Özyurt and C. Köse. Chat mining: Automatically determination of chat conversations' topic in Turkish text based Chat mediums. *Expert Syst. Appl.*, 37:8705–8710, 2010.

[81] C. D. Paice. Another stemmer. *SIGIR Forum*, 24(3):56–61, 1990.

[82] J. Pearl. Bayesian networks: A model of self-activated memory for evidential reasoning. In *Proc. of the 7th Conference of the Cognitive Science Society*, pages 329–334, 1985.

[83] N. Pendar. Toward spotting the pedophile telling victim from predator in text chats. In *Proc. of the International Conference on Semantic Computing*, pages 235–241, Washington, DC, USA, 2007. IEEE Computer Society.

[84] J. C. Platt. Fast Training of Support Vector Machines Using Sequential Minimal Optimization. pages 185–208, 1999.

[85] A. Pons-Porrata, R. Berlanga-Llavori, and J. Ruiz-Shulcloper. Topic discovery based on text mining techniques. *Inf. Process. Manage.*, 43(3):752–768, 2007.

[86] M. F. Porter. An algorithm for suffix stripping. *Program*, 3(14):130–137, October 1980.

[87] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[88] J. R. Quinlan. Learning with continuous classes. In *Proc. of the 5th Australian Joint Conference on Artificial Intelligence*, pages 343–348. World Scientific, 1992.

[89] J. R. Quinlan. C4.5: Programs for machine learning. In *Machine Learning*, pages 343–348. Morgan Kaufmann Publishers, San Mateo, CA., 1993.

[90] V. D. H. Renee. Introduction to social network analysis (sna) as an investigative tool. *Trends in Organized Crime*, 12:101–121, 2009.

[91] S. E. Robertson and Sparck K. J. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.

[92] J. Rudman. The state of authorship attribution studies: Some problems and solutions. *Computers and the Humanities*, 31:351–365., 1998.

[93] G. Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[94] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[95] J. Schroeder, J. J. Xu, H. Chen, and M. Chau. Automated criminal link analysis based on domain knowledge. *Journal of the American Society for Information Science and Technology*, 58(6):842–855, 2007.

[96] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[97] D. K. Sepandar, D. Klein, and D. M. Christopher. Interpreting and extending classical agglomerative clustering algorithms using a model-based approach. In *Proc. ICML*, 2002.

[98] M. Sewell. Feature selection, 2007. http://machine-learning.martinsewell.com /feature-selection/feature-selection.pdf.

[99] J. Shetty. Discovering important nodes through graph entropy: The case of enron email database. In *Proc. of the 3rd international workshop on Link discovery*, pages 74–81. Press, 2005.

[100] E. Stamatatos, G. Kokkinakis, and N. Fakotakis. Automatic text categorization in terms of genre and author. *Computational Linguistics*, 26(4):471–495, 2000.

[101] S. J. Stolfo, G. Creamer, and S. Hershkop. A temporal based forensic analysis of electronic communication. In *Proc. of the International Conference on Digital Government Research*, pages 23–24, San Diego, California, 2006. ACM.

[102] S. J. Stolfo and S. Hershkop. Email mining toolkit supporting law enforcement forensic analyses. In *Proc. of the national conference on digital government research*, pages 221–222. Digital Government Society of North America, 2005.

[103] J. Su, H. Zhang, C. X. Ling, and S. Matwin. Discriminative Parameter Learning for Bayesian Networks. In *Proc. of the International Conference on Machine Learning*, pages 1016–1023, 2008.

[104] G. Teng, M. Lai, J. Ma, and Y. Li. E-mail authorship mining based on svm for computer forensic. In *Proc. of the 3rd International Conference on Machine Learning and Cyhemetics*, Shanghai, China, 2004.

[105] R. Thomson and T. Murachver. Predicting gender from electronic discourse. *British Journal of Social Psychology*, 40(2):193–208, 2001.

[106] F. J. Tweedie and R. H. Baayen. How variable may a constant be? measures of lexical richness in perspective. *Computers and the Humanities*, 32:323–352, 1998.

[107] H. van Haltern. Author verification by linguistic profiling: An exploration of the parameter space. *ACM Transactions on Speech and Language Processing*, 4(1), 2007.

[108] G. Wang, H. Chen, and H. Atabakhsh. Automatically detecting deceptive criminal identities. *Commun. ACM*, 47(3):70–76, 2004.

[109] M. White, T. Korelsky, C. Cardie, V. Ng, D. Pierce, and K. Wagstaff. Multidocument summarization via information extraction. In *Proc. of the first international conference on Human language technology research*, pages 1–7, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[110] K. Wimmer. The First Amendment and the Media, 2002. http://www.mediainstitute.org/ONLINE/FAM2002/toc.html.

[111] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*. Elsevier, June, 2005.

[112] Y. Xiang, M. Chau, H. Atabakhsh, and H. Chen. Visualizing criminal relationships: comparison of a hyperbolic tree and a hierarchical list. *Decis. Support Syst.*, 41(1):69–83, 2005.

[113] R. Xiong and J. Donath. Peoplegarden: Creating data portraits for users. In *Proc. of the 12th annual ACM symposium on User Interface Software and Technology (UIST '99). ACM Press*, pages 37–44. ACM, 1999.

[114] V. A. Yatsko and T. N. Vishnyakov. A method for evaluating modern systems of automatic text summarization. *Automatic Documentation and Mathematical Linguistics*, 41(3):93–103, 2007.

[115] F. Yoav and R. E. Schapire. Experiments with a New Boosting Algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[116] G. U. Yule. On sentence length as a statistical characteristic of style in prose. *Biometrika*, 30:363–390, 1938.

[117] G. U. Yule. *The statistical study of literary vocabulary*. Cambridge University Press, Cambridge, UK, 1944.

[118] M. J. Zaki. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 12:372–390, 2000.

[119] Y. Zhang, N. Zincir-Heywood, and E. Milios. Narrative text classification for automatic key phrase extraction in web document corpora. In *WIDM '05: Proc. of*

the 7th annual ACM international workshop on Web information and data management, pages 51–58, New York, NY, USA, 2005. ACM.

[120] Y. Zhao and J. Zobel. Effective and scalable authorship attribution using function words. In *Proc. of the Second AIRS Asian Information Retrieval Symposium*, pages 174–189. Springer, 2005.

[121] R. Zheng, J. Li, H. Chen, and Z. Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American Society for Information Science and Technology*, 57(3):1532–2882, 2006.

[122] R. Zheng, Y. Qin, Z. Huang, and H. Chen. Authorship analysis in cybercrime investigation. In *Proc. of the 1st NSF/NIJ Symposium, ISI2003*, pages 59–73. Springer-Verlag, 2003.

Appendices

Appendix I: Function Words

a	an	at	as	above	are	about
because	be	in	some	nor	but	us
including	both	upon	inside	of	used	someone
we	they	their	that	by	into	off
although	once	than	what	do	one	its
the	when	each	opposite	less	where	and
or	whether	little	our	these	which	every
whom	everyday	many	this	anyone	who	everything
most	my	per	from	whoever	with	past
few	must	though	plus	following	own	those
him	no	should	unlike	yes	below	nobody
if	none	so	up	your	regarding	toward
without	need	several	under	worth	before	her
between	such	is	on	latter	onto	enough
anybody	over	more	following	around	plenty	for
after	am	any	nothing	somebody	all	he
she	it	something	via	can	beside	behind
whatever	among	down	either	like	them	would
lots	outside	while	will	till	through	to
whose	me	everyone	much	anything	same	towards
since	until	you	within	have	near	neither
unless	i	another				

Appendix II: Gender-specific Features

1. No. of words ending with able / W
2. No. of words ending with al / W
3. No. of words ending with ful / W
4. No. of words ending with ible / W
5. No. of words ending with ic / W
6. No. of words ending with ive / W
7. No. of words ending with less / W
8. No. of words ending with ly / W
9. No. of words ending with ous / W
10. No. of occurrences of ‘sorry’ / W
11. No. of occurrences of ‘apology’ / W

    where the letter ‘W’ denotes the total number of words or tokens.