A Novel Approach of Mining Write-Prints for Authorship Attribution in E-mail Forensics

Farkhund Iqbal Rachid Hadjidj Benjamin C. M. Fung Mourad Debbabi
Concordia Institute for Information Systems Engineering
Faculty of Engineering and Computer Science
Concorida University
Montreal, Quebec
Canada H3G 1M8
{iqbal_f, hadjidj}@encs.concordia.ca {fung, debbabi}@ciise.concordia.ca

ABSTRACT

There is an alarming increase in the number of cyber-crime incidents through anonymous e-mails. The problem of e-mail authorship attribution is to identify the most plausible author of an anonymous e-mail from a group of potential suspects. Most previous contributions employed a traditional classification approach, such as decision tree and Support Vector Machine (SVM), to identify the author and studied the effects of different writing style features on the classification accuracy. However, little attention has been given on ensuring the quality of the evidence. In this paper, we introduce an innovative data mining method to capture the write-print of every suspect and model it as *combinations of features* that occurred frequently in the suspect's emails. This notion is called *frequent pattern*, which has proven to be effective in many data mining applications, but it is the first time to be applied to the problem of authorship attribution. Unlike the traditional approach, the extracted write-print by our method is *unique* among the suspects and, therefore, provides convincing and credible evidence for presenting it in a court of law. Experiments on real-life e-mails suggest that the proposed method can effectively identify the author and the results are supported by a strong evidence.

Keywords

E-mail forensic analysis, authorship identification, data mining, write-print, frequent itemsets

1. INTRODUCTION

E-mail is one of the most widely used way of written communication over the Internet, and its traffic has increased exponentially with the advent of world wide web. Trillions of business letters, financial transactions, governmental orders and friendly messages are exchanged through e-mail system each year. The increase in e-mail traffic comes also with an increase in the use of e-mails for illegitimate purposes [18]. Phishing, spamming, e-mail bombing, threatening, cyber bullying, racial vilification, child pornography, and sexual harassments are common examples of e-mail abuses. Terrorist groups and criminal gangs are using e-mail systems as a safe channel for their communication. E-mail is also abused for committing infrastructure crimes by transmitting worms, viruses, trojan horses, hoaxes and other malicious executables over the Internet. In many misuse cases, the criminals attempt to hide their true identity. Likewise, in phishing, a person may try to impersonate a manager or a financial adviser to obtain clients' secret information.

E-mail systems are inherently vulnerable to misuse for three main reasons. First, an e-mail can be spoofed and the meta data contained in its header about the sender and the path along which the message has travelled can be forged or anonymized. An e-mail can be routed through anonymous e-mail servers to hide the information about its origin. Second, e-mail systems are capable of transporting executables, hyperlinks, trojan horses, and scripts. Third, the Internet including e-mail services are accessible through public places, such as net cafes and libraries, which further deteriorates the anonymity issue. Presently, there is no adequate proactive mechanism to prevent e-mail misuses, and merely installing filters and firewalls are insufficient. In this situation, forensic e-mail analysis with special focus on authorship attribution can help prosecute the offender of e-mail misuse by means of law [18].

The *problem of authorship attribution* in the context of e-mail forensics can be described as follows: A cyber forensic investigator wants to determine the author of a given malicious e-mail μ and has to identify 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. In forensic science, an individual can be uniquely identified by his/her fingerprint. Similarly, in cyber forensics, an investigator would like to identify the "write-print" of an individual from his/her e-mails and use it for au-

thorship attribution. The key question is:

What exactly are the patterns that can represent the write-print of an individual?

Our insight is that the write-print of an individual is the *combinations of features* that occur frequently in his/her written e-mails. The commonly used features are lexical, syntactical, structural and content-specific attributes (see Section 2.1). By matching the write-print with the malicious e-mail, the true author can be identified. Most importantly, the matched write-print *should* provide credible evidence for supporting the conclusion. The research community [6][18][23] has devoted a lot of efforts in studying stylistic and structural features *individually*, but very few of them has studied the *combinations* of features that form a write-print and addressed the issue of evidence gathering.

The classification models employed in previous contributions on authorship attribution has two broad categories: Decision tree (C4.5) [17] and Support Vector Machine (SVM) [5]. 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 problem by considering all features when a hyperplane is created. However, SVM is a like a blackbox function which takes some input (the malicious e-mail) and provides an output (the author). It fails to provide intuitive explanation of how it arrives to a certain conclusion. Therefore, SVM may not be the best choice in the context of e-mail forensic investigation, where collecting credible evidence is one of the primary objectives.

In this paper, we are introducing a novel approach of authorship attribution in which the unique write-print of every suspect is extracted. These write-prints are used to identify the true author of a disputed e-mail, and to gather convincing and credible evidence to support the finding. To concisely model the write-print of an individual, we borrow the concept of *frequent pattern* (a.k.a. *frequent itemset*) [2] from data mining to capture the combinations of features that frequently occurred in an individual's e-mails. 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 many other applications of pattern recognition. To the best of our knowledge, this is the first paper introducing the concept of frequent pattern to the problem of authorship attribution.

Figure 1 depicts an overview of our proposed method. We first extract the set of frequent patterns independently from the e-mails E_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 write-print of a suspect S_i

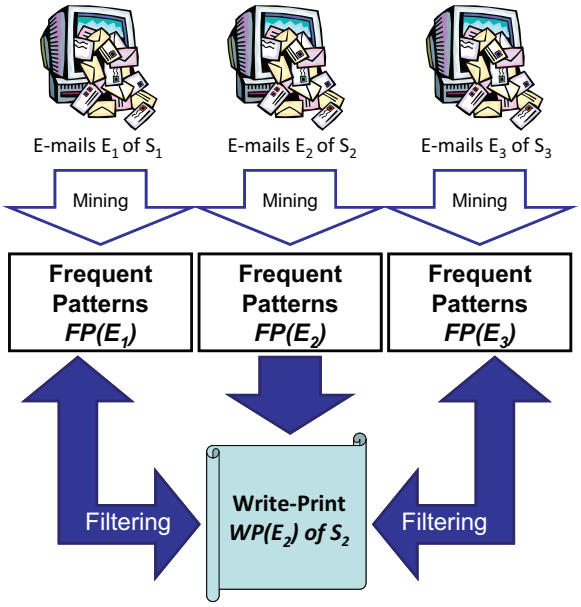


Figure 1: Mining write-print $WP(E_2)$ of S_2

because an other 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 *write-print* of a suspect. 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 e-mails of a particular suspect. Our method guarantees that the identified patterns are frequent in the e-mails of one suspect only, and are not frequent in others' emails. It will be difficult for the accused suspect to deny the validity of the findings. The results obtained are traceable, justifiable, and can be presented quantitatively with a statistical support.

- *Flexible writing styles:* The frequent patterns mining technique can adopt all four types of commonly used writing style features (described in Section 2.1). 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 the traditional approaches where it is hard to determine the contribution of each feature in the authorship attribution process [7], the proposed technique is based

on the distinctive patterns, which are the combination of features. The support associated to each pattern in the write-print set determines the contribution of each pattern.

- *Generic application:* The dataset used in most of the existing techniques are constrained by the number, size and topic of e-mails. Our experiments on the real-life data, the Enron e-mail corpus, suggest that the proposed approach is very robust to these factors. This is crucial for the application in real world investigations.

The rest of the paper is organized as follows. Section 2 reviews the previous contributions. Section 3 formally defines the problem and the notions of write-print. Section 4 describes our proposed approach. Section 5 evaluates our proposed method on real-life e-mail dataset. Section 6 concludes the paper.

2. RELATED WORK

Most previous contributions on authorship attribution are applications of text classification analysis [6]. The process starts by identifying a set of writing style features of a person that are relatively common in most of his works. A classifier is trained on the collected writing style features to build a model, which is then used to classify the disputed e-mail to the most plausible author among the suspects. In this section, we review the commonly employed writing style features and summarize the techniques of e-mail authorship attribution, found in the literature of authorship attribution.

2.1 Writing Style Features

There is no pre-defined set of features that can be used to differentiate the writing styles of different suspects. The writing patterns usually contain the characteristics of words usage, words sequence, composition and layouts, common spelling and grammatical mistakes, vocabulary richness, hyphenation and punctuation. Abbasi and Chen [1] presented a comprehensive analysis on the stylistics features. Below, we provide a summary of the common writing style features, namely, lexical, syntactical, structural and content-specific attributes.

Lexical Features are the characteristics of both characters and words or tokens. In terms of characters, for instance, frequency of letters, frequency of capital letters, total number of characters per token and character count per sentence, are the most relevant metrics. Word-based lexical features may include word lengths distribution, words per sentence, and vocabulary richness. Initially, researchers thought that vocabulary richness [20][21] and word usage [12] are discriminating features to be used for authorship attribution. *Syntactic Features* include the distribution of function words (such as "upon", "thus", "above") and punctuation play a significant role in authorship attribution [3][13][19]. *Structural Features* are used to measure the overall layout and organization of text within documents. For instance, average paragraph length, number of paragraphs per document, presence of greetings and their position within the e-mail are common structural features. Moreover, the presence of sender signature including his contact information is one of the special structural feature of e-mail documents. *Content-specific Features* are collection of certain key-words commonly found in a specific domain and may vary from context to context even for the same author. Zheng et al. [23] used 11 keywords from the cybercrime taxonomy in authorship analysis experiments.

2.2 E-mail Authorship Analysis

Authorship analysis has been very successful in resolving authorship attribution disputes over literary and conventional writings [16]. However, e-mail authorship attribution poses some special challenges due to its special characteristics of size and composition, as compared to literary works [7]. Literary documents are usually large in size comprising of (at least) several paragraphs and have a definite syntactic and semantic structure. In contrast, e-mails are short in size and usually do not follow definite syntactic or grammatical rules, therefore, it is hard to learn from them about the writing patterns of their author. Ledger and Merriam [15], for instance, established that authorship analysis results would not be significant for texts containing less than 500 words. Moreover, e-mails are more interactive and informal in style, and people are not couscous about the spelling and grammatical mistakes particularly in informal e-mails. Therefore, techniques which are very successful in literary and traditional works are not applicable in the e-mail authorship attribution.

Teng et al. [18] and De Vel et al. [6] applied support vector machine (SVM) classification model over a set of stylistic and structural features for e-mail authorship attribution. De Vel et al. [8] and Corney et al. [4] performed extensive experiments and found that the classification accuracy decreases when the size of training set decreases, the number of authors increases, or the length of documents decreases. Recently, Zheng et al. [23][24] used a comprehensive set of lexical, syntactical and structural features including 10-11 content-specific keywords. Van Halteren [10] used the same set of linguistic features for authorship attribution of class essays. De Vel et al. [6] further found that by increasing the number of function words from 122 to 320, the performance of SVM worsened, which weakens the argument that SVM supports high dimensionality. This result also illustrates that adding more features does not necessarily improve the accuracy. In contrast, in

this paper we focus on identifying the combinations of key features that can differentiate the writing style of different suspects and filtering out the useless features that do not contribute towards the goal of authorship attribution.

In the current literature, each type of the four features sets is applied independently from the other, which may otherwise produce different results [6]. For instance the word usage and composition style may vary from one structural pattern to another. In our approach, the write-prints could be the combination of all the four types of writing style features. Moreover, the current literature of authorship attribution suffers from the problem of having too many features. It is difficult to determine the set of features that should be used for a given set of e-mails. Previous research [6] has shown that adding useless features may decrease the accuracy because a classifier may capture the useless features as noise. Using those noisy features for classification diminishes the justification of evidence for supporting the finding. Some studies identify some particular useful style markers, but the identified style markers are data dependent and may not be applicable to other data sets. Our approach overcomes this limitation by flexibly extracting the evidence (combinations of frequently occurred features) from the data itself, provided. The insignificant noisy features are filtered out.

3. PROBLEM STATEMENT

Let $\{S_1, \cdots, S_n\}$ be the set of suspected authors of a malicious e-mail μ. We assume that there is a collection of e-mails, denoted by E_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 e-mails E_a has the "best match" with the patterns in the malicious e-mail μ. Intuitively, a collection of e-mails E_i *matches* μ if E_i and μ share similar patterns of vocabulary usage, structural and/or stylometric features. The primary objective of a cyber forensic investigator is to precisely extract the patterns of each suspect, so she can use such patterns to identify the author of the malicious e-mail μ and present such patterns as evidence to support her finding.

What are the patterns that can represent the "write-print" of a suspect S_i? Specifically, we want to extract the patterns that *uniquely* represent the writing style of a suspect S_i, but does not represent the writing style of *any* other suspect S_j, where $i \neq j$. In the rest of this section, we discuss the pre-processing of features and formally define the notions of *frequent pattern* and *write-print*.

3.1 Pre-Processing

Let E_i be a collection of e-mails written by suspect $S_i \in \{S_1, \cdots, S_n\}$. First, we extract the features from each e-mail in E_i. In the rest of this section, the term "feature" refers to either a stylometric feature described in Section 2.1 or a word appearing in the e-mails. The spaces, punctuation, special characters and blank lines are removed. Next, we discretize each normalized word frequency into a set of intervals, for example, [0-0.25], (0.25-0.5], (0.5-0.75], (0.75-1]. Each interval is called a *feature item*. The normalized feature frequency is then matched with these intervals. Then assign value 1 to the feature item if the interval contains the normalized feature frequency; otherwise assign value 0. This will simplify the procedure by determining the presence or absence of a pattern. 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.

EXAMPLE 3.1. Consider Table 1, which contains 10 e-mails. We extracted three features $\{A, B, C\}$ from the 10 e-mails. We first discretize each feature into feature items. For example, a stylometric feature A having a normalized range of $[0, 1]$ can be discretized into four intervals $A1 = [0, 0.25]$, $A2 = (0.25, 0.5]$, $A3 = (0.5, 0.75]$, $A4 = (0.75, 1]$, representing four feature items. Similarly, features B and C are discretized into $B1 = [0, 0.5]$, $B2 = (0.5, 1]$, $C1 = [0, 0.5]$, and $C2 = (0.5, 1]$. An e-mail ε_1 having features $A = 0.3$, $B = 0.25$, and $C = 0.25$ can be represented as feature vector $\langle$ 0, 1, 0, 0, 1, 0, 1, 0 $\rangle$. ■

3.2 Frequent Pattern

Intuitively, the "writing pattern" or the "writing style" in an ensemble of e-mails E_i (written by suspect S_i) is a combination of feature items that *frequently* occurs in E_i. We concisely model and capture such frequently occurred patterns by the concept of *frequent itemset* [2] described as follows.

Let $U = \{f_1, \cdots, f_m\}$ denote the universe of all feature items. Let E_i be a set of e-mails where each e-mail ε is represented as a set of feature items such that $\varepsilon \subseteq U$. An e-mail ε contains a feature item f_i if the numerical feature value of the e-mail ε falls within the interval of f_i. For example, e-mail ε_1 in Table 1 can be represented as a set of feature items $\varepsilon_1 = \{A2, B1, C1\}$. Table 2 shows the 10 e-mails from Table 1 in this format.

Let $F \subseteq U$ be a set of feature items called an *pattern*. An e-mail ε *contains* a pattern F if $F \subseteq \varepsilon$. A pattern that contains k feature items is a *k-pattern*. For

	Feature A				Feature B		Feature C	
E-mail	**A1**	**A2**	**A3**	**A4**	**B1**	**B2**	**C1**	**C2**
ε_1	0	1	0	0	1	0	1	0
ε_2	0	1	0	0	1	0	1	0
ε_3	0	1	0	0	1	0	1	0
ε_4	1	0	0	0	1	0	1	0
ε_5	0	0	0	1	1	0	1	0
ε_6	0	0	1	0	0	1	0	1
ε_7	0	0	0	1	1	0	0	1
ε_8	0	0	1	0	0	1	0	1
ε_9	0	1	0	0	1	0	0	1
ε_{10}	1	0	0	0	1	0	0	1

Table 1: Feature Vectors

E-mail
$\varepsilon_1 = \{A2, B1, C1\}$
$\varepsilon_2 = \{A2, B1, C1\}$
$\varepsilon_3 = \{A2, B1, C1\}$
$\varepsilon_4 = \{A1, B1, C1\}$
$\varepsilon_5 = \{A4, B1, C1\}$
$\varepsilon_6 = \{A3, B2, C2\}$
$\varepsilon_7 = \{A4, B1, C2\}$
$\varepsilon_8 = \{A3, B2, C2\}$
$\varepsilon_9 = \{A2, B1, C2\}$
$\varepsilon_{10} = \{A1, B1, C2\}$

Table 2: Feature Items

example, the pattern $F = \{f_1, f_4, f_6\}$ is a 3-pattern. The *support* of a pattern F is the percentage of e-mails in E_i that contains F. A pattern F is a *frequent pattern* in a set of e-mails E_i if the support of F is greater than or equal to some user-specified minimum support threshold.

Definition 3.1 (Frequent pattern). Let E_i be the set of e-mails written by suspect S_i. Let $support(F|E_i)$ be the percentage of e-mails in E_i that contain the pattern F, where $F \subseteq U$. A pattern F is a *frequent pattern* in E_i if $support(F|E_i) \geq min\_sup$, where the minimum support threshold min_sup is a real number in an interval of $[0, 1]$. ■

The writing pattern of a suspect S_i is represented as a set of frequent patterns, denoted by $FP(E_i) = \{F_1, \cdots, F_k\}$, extracted from his/her e-mails E_i.

Example 3.2. Consider Table 2. Suppose the user-specified threshold $min\_sup = 0.3$, which means that a pattern $F = \{f_1, \cdots, f_k\}$ is frequent if at least 3 out of the 10 e-mails contain all feature items in F. $\{A1\}$ is not a frequent pattern because it has support 2/10=0.2. $\{A2\}$ is a 1-frequent pattern because it has support 0.4. $\{A2, B1\}$ is a 2-frequent pattern because it has support 0.4. $\{A2, B1, C1\}$ is a 3-frequent pattern because it has support 0.3. Example 4.1 will show how to efficiently compute all frequent patterns. ■

3.3 Write-Print

In forensic science, an individual can be uniquely identified by his/her fingerprint. In cyber forensics, can we identify the "write-print" of an individual from his/her e-mails? We do not claim that the identified write-print in this paper can uniquely distinguish every individual in the world, but the identified write-print 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 write-print.

The notion of frequent patterns in Definition 3.1 captures the writing pattern 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 write-print.

Intuitively, a write-print can uniquely represent the writing style of a suspect S_i if its pattern is found *only* in the e-mails written by S_i, but not in any other suspect's e-mails. In other words, the write-print of a suspect S_i is a pattern F that is frequent in the emails E_i written by S_i but not frequent in the e-mails E_j written by any other suspect S_j where $i \neq j$.

Definition 3.2 (Write-print). A *write-print*, denoted by $WP(E_i)$, is a set of patterns where each pattern F has $support(F|E_i) \geq min\_sup$ and $support(F|E_j) < min\_sup$ for any E_j where $i \neq j$, min_sup is a user-specified minimum threshold. In other words, $WP(E_i) \subseteq FP(E_i)$, and $WP(E_i) \cap WP(E_j) = \emptyset$ for any $1 \leq i, j \leq n$ and $i \neq j$. ■

Discussion: Our notion of write-print has two special

properties that make it different from the traditional notion of write-print in the literature.

First, the *combination* of feature items that composes the write-print of a suspect is dynamically generated based on the embedded pattern in the e-mails. This flexibility allows us to succinctly model the write-print of different suspects by using different combinations of feature items. In contrast, the traditional notion of write-print considers one feature at a time without considering the combinations.

Second, every frequent pattern F in our notion of write-print captures a piece of writing pattern that can be found *only* in one suspect's emails, but not in any other suspects' e-mails. The cyber forensic investigator could precisely point out such matched patterns in the malicious e-mail to support her conclusion of authorship identification. In contrast, the traditional classifier, e.g., decision tree, attempts to use the *same* set of features to capture the write-print of different suspects. It is quite possible that the classifier would capture some common writing patterns and the investigator could unintentionally use those common patterns to draw the wrong conclusion of authorship. Our notion of write-print avoids such problem and, therefore, provides more convincing and reliable evidence.

3.4 Refined Problem Statement

The problem of authorship attribution can be refined into three subproblems: (1) To identify the write-print $WP(E_i)$ from each set of e-mails $E_i \in \{E_1, \cdots, E_m\}$. (2) To determine the author of the malicious e-mail μ by matching μ with each of $\{WP(E_1), \cdots, WP(E_m)\}$. (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 can first extract the set of frequent patterns $FP(E_i)$ from E_i and then filter out the common frequent patterns that also appear in any other sets of emails E_j. For subproblem (3), the write-print $WP(E_a)$ could serve the evidence for supporting the conclusion, where E_a is the set of e-mails written by the identified author S_a.

4. OUR METHOD

Algorithm 1 presents a novel data mining method, called *AuthorMiner*, for determining the authorship of a malicious e-mail μ from a group of suspects $\{S_1, \cdots, S_n\}$ based on the extracted features of their previously written e-mails $\{E_i, \cdots, E_n\}$. In this section, an e-mail is represented by a set of feature items. Below, we summarize the algorithm in three phases. Sections 4.1-4.3 discuss each phase in detail.

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

Phase 2: Filtering common frequent patterns (Lines 4-16). Though $FP(E_i)$ has captured the writing patterns of suspect S_i, $FP(E_i)$ may contain frequent patterns that are common to other suspects. Therefore, Phase 2 is to remove the common frequent patterns. Specifically, a frequent pattern F in $FP(E_i)$ is removed if *any* other $FP(E_j)$ also contains F, where $i \neq j$. The remaining frequent patterns in $FP(E_i)$ form the write-print $WP(E_i)$ of suspect S_i. When this phase completes, we have a set of write-prints $\{WP(E_1), \cdots, WP(E_n)\}$ of suspects $\{S_1, \cdots, S_n\}$. Figure 1 illustrates that the write-print $WP(E_2)$ comes from $FP(E_2)$ and filters out the common patterns by comparing with $FP(E_1)$ and $FP(E_3)$.

Phase 3: Identifying author (Lines 17-24). Compare the malicious e-mail μ with each write-print $WP(E_i) \in \{WP(E_1), \cdots, WP(E_n)\}$ and identify the most similar write-print that matches μ. Intuitively, a write-print

Algorithm 1 AuthorMiner

Require: The malicious e-mail μ.
Require: A set of e-mail $\{E_1, \cdots, E_n\}$, written by $\{S_1, \cdots, S_n\}$.
/* Mining frequent patterns */
1: **for each** $E_i \in \{E_1, \cdots, E_n\}$ **do**
2: extract frequent patterns $FP(E_i)$ from E_i;
3: **end for**
/* Filtering out common frequent patterns */
4: **for each** $FP(E_i) \in \{FP(E_1), \cdots, FP(E_n)\}$ **do**
5: **for each** $FP(E_j) \in \{FP(E_{i+1}), \cdots, FP(E_n)\}$ **do**
6: **for each** frequent pattern $F_x \in FP(E_i)$ **do**
7: **for each** frequent pattern $F_y \in FP(E_j)$ **do**
8: **if** $F_x == F_y$ **then**
9: $FP(E_i) = FP(E_i) - F_x$;
10: $FP(E_j) = FP(E_j) - F_y$;
11: **end if**
12: **end for**
13: **end for**
14: **end for**
15: $WP(E_i) = pattern(E_i)$;
16: **end for**
/* Identifying author */
17: $highest\_score = -1.0$;
18: **for all** $WP(E_i) \in \{WP(E_1), \cdots, WP(E_n)\}$ **do**
19: **if** $Score(\mu \approx WP(E_i)) > highest\_score$ **then**
20: $highest\_score = Score(\mu \approx WP(E_i))$;
21: $author = S_i$;
22: **end if**
23: **end for**
24: **return** $author$;

$WP(E_i)$ is similar to the e-mail μ if many frequent patterns in $WP(E_i)$ can be found in μ. Our insight is that the frequent patterns are not equally important. Their importance is reflected by their $supprt(F|E_i)$; therefore, we derive a score function, $Score(\mu \approx WP(E_i))$ to measure the weighted similarity between the e-mail μ and the frequent patterns in $WP(E_i)$. The suspect S_a of write-print $WP(E_a)$, which has the highest $Score(\mu \approx WP(E_i))$, is classified to be the author of the malicious e-mail μ.

4.1 Mining Patterns (Lines 1-3)

Lines 1-3 mine the frequent patterns $FP(E_i)$ from each collection of e-mail $E_i \in \{E_1, \cdots, E_n\}$, for $1 \leq i \leq n\}$. There are many data mining algorithms for extracting frequent patterns, for example, Apriori [2], FP-growth [11], and ECLAT [22]. Below, we provide an overview of the Apriori algorithm which has been previously applied to various text mining tasks [9][14].

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

PROPERTY 4.1 (APRIORI PROPERTY). All nonempty subsets of a frequent pattern must also be frequent. ■

By definition, a pattern F' is not frequent if $support(F'|E_i) < min\_sup$. The above property implies that adding a feature item f to a non-frequent pattern F' will never make it more frequent. Thus, if a k-pattern F' is not frequent, then there is no need to generate $(k+1)$-pattern $F' \cup f$ because $F' \cup f$ is also not frequent. The following example shows how the Apriori algorithm exploits this property to efficiently extract all frequent patterns. Refer to [2] for a formal description.

EXAMPLE 4.1. Consider Table 2 with $min\_sup = 0.3$. First, identify all frequent 1-patterns by scanning the database once to obtain the support of every item. The items having support ≥ 0.3 are frequent 1-patterns, denoted by $L_1 = \{\{A2\}, \{B1\}, \{C1\}, \{C2\}\}$. Then, join L_1 with itself, i.e. $L_1 \bowtie L_1$, to generate the candidate set $C_2 = \{\{A2, B1\}, \{A2, C1\}, \{A2, C2\}, \{B1, C1\}, \{B1, C2\}, \{C1, C2\}\}$ and scan the database once to obtain the support of every pattern in C_2. Identify the frequent 2-patterns, denoted by $L_2 = \{\{A2, B1\}, \{A2, C1\}, \{B1, C1\}, \{B1, C2\}\}$. Similarly, perform $L_2 \bowtie L_2$ to generate C_3 scan the database once to identify the frequent 3-pattern which is $L_3 = \{\{A2, B1, C1\}\}$. The finding of each set of frequent k-patterns requires one full scan of the e-mail feature items in Table 2. ■

4.2 Filtering Common Patterns (Lines 4-16)

This phase filters out the common frequent patterns among $\{FP(E_1), \cdots, FP(E_n)\}$. Lines 4-16 in Algorithm 1 describe the filtering procedure. The general idea is to compare every frequent pattern F_x in $FP(E_i)$ with every frequent pattern F_y in *all* other $FP(E_j)$, and to remove them from $FP(E_i)$ and $FP(E_j)$ if F_x and F_y are the same. The computational complexity of this step is $O(|FP(E)|^n)$ where $|FP(E)|$ is the number of frequent patterns in $FP(E)$ and n is the number of suspects. The remaining frequent patterns in $FP(E_i)$ form the write-print $WP(E_i)$ of suspect S_i.

EXAMPLE 4.2. Suppose there are three suspects S_1, S_2, and S_3 having three sets of e-mails E_1, E_2, and E_3 respectively, as depicted in Figure 1. Let $FP(E_1) = \{\{A1\}, \{B1\}, \{C2\}, \{A1, B1\}, \{A1, C2\}, \{B1, C2\}, \{A1, B1, C2\}\}$ be the frequent patterns of S_1. Let $FP(E_2) = \{\{A2\}, \{B1\}, \{C1\}, \{C2\}, \{A2, B1\}, \{A2, C1\}, \{B1, C1\}, \{B1, C2\}, \{A2, B1, C1\}\}$ be the set of frequent patterns from Example 4.1 of S_2. Let $FP(E_3) = \{\{A1\}, \{B3\}, \{C2\}, \{A1, B3\}, \{A1, C2\}, \{B3, C2\}, \{A1, B3, C2\}\}$ be the set of frequent patterns of S_3. Then, we discard $\{A1\}, \{B1\}, \{C2\}, \{A1, C2\}, \{B1, C2\}$ because more than one set of frequent patterns contains them. The remaining frequent patterns form the write-print of the suspect: $WP(E_1) = \{\{A1, B1\}, \{A1, B1, C2\}\}$, $WP(E_2) = \{\{A2\}, \{C1\}, \{A2, B1\}, \{A2, C1\}, \{B1, C1\}, \{A2, B1, C1\}\}$, and $WP(E_3) = \{\{B3\}, \{A1, B3\}, \{B3, C2\}, \{A1, B3, C2\}\}$. ■

4.3 Identifying Author (Lines 17-24)

Lines 17-24 determine the author of the malicious e-mail μ by comparing μ with each write-print $WP(E_i) \in \{WP(E_1), \cdots, WP(E_n)\}$ and identifying the most similar write-print to μ. Intuitively, a write-print $WP(E_i)$ is similar to μ if many frequent patterns in $WP(E_i)$ matches the style in μ. Formally, a frequent pattern F *matches* μ if μ contains every feature item in F.

Equation 1 shows the score function that quantifies the similarity between the malicious e-mail μ and a write-print $WP(E_i)$. The frequent patterns are not equally important, and their importance is reflected by their support in E_i, i.e., the percentage of e-mails in E_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 $WP(E_i)$ to normalize the factor of different sized $WP(E_i)$.

$$Score(\mu \approx WP(E_i)) = \frac{\sum_{j=1}^{p} support(MP_j|E_i)}{|WP(E_i)|} \quad (1)$$

where $MP = \{MP_1, \cdots, MP_p\}$ is a set of matched patterns between $WP(E_i)$ and the malicious e-mail μ. The score is a real number within the range of $[0, 1]$. The higher the score means the higher similarity between the write-print and the malicious e-mail μ. The suspect having the write-print with the highest score is the author of the malicious e-mail μ.

EXAMPLE 4.3. Let the patterns found in the malicious e-mail μ be $\{A2, B1, C1\}$ and $\{A1, B1, C2\}$. Comparing them to the write-prints in Example 4.2, we notice that the first pattern matches to a pattern in $WP(E_2)$ while the second pattern matches to a pattern in $WP(E_1)$. The score calculated according to Equation 1 is higher for $WP(E_1)$ because $|WP(E_1)| < |WP(E_2)|$. As a result, the malicious e-mail μ is most similar to $WP(E_1)$, suggesting that S_1 is the author. ■

In an unlikely case where multiple suspects have the same highest score, we return all of them to the user.

5. EXPERIMENTAL EVALUATION

Our goals in this section are to evaluate the proposed method, AuthorMiner, in terms of authorship identification accuracy and to verify the extracted write-print exhibits strong evidence for supporting the conclusion on authorship. We employed the Enron E-mail Dataset[1], which contains 200,399 real-life e-mails from 158 employees of the Enron corporation after cleaning. As a pre-processing step, we removed the empty spaces, special characters, and blank lines and tokenized the e-mails as described in Section 3.1. Unlike the ordinary text mining application which aims at extracting the general trends in the text, our goal is to differentiate the writing style of different suspects. Therefore, we keep all the function words and short words.

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-mails are for training and the remaining $\frac{1}{3}$ of the m e-mails are for testing. We then applied our method, AuthorMiner, to extract the write-prints from $\{S_1, \cdots, S_n\}$ 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.

Figure 2 depicts the authorship identification accuracy for $n = 6$ and $m = 20$ (i.e., a total of 120 e-mails) on different number of discretized intervals. The accuracy spans from 86% to 90% at $min\_sup = 0.1$, 0.3 and 0.5, suggesting that our proposed method can effectively identify the author of an e-mail based on the extracted write-prints when a reasonable min_sup is specified. As min_sup increases, the number of extracted frequent patterns, i.e. $|FP(E_i)|$, 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 write-print becomes less effective for authorship identification and the accuracy decreases.

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

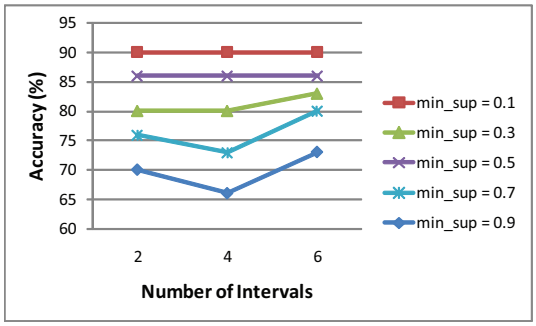


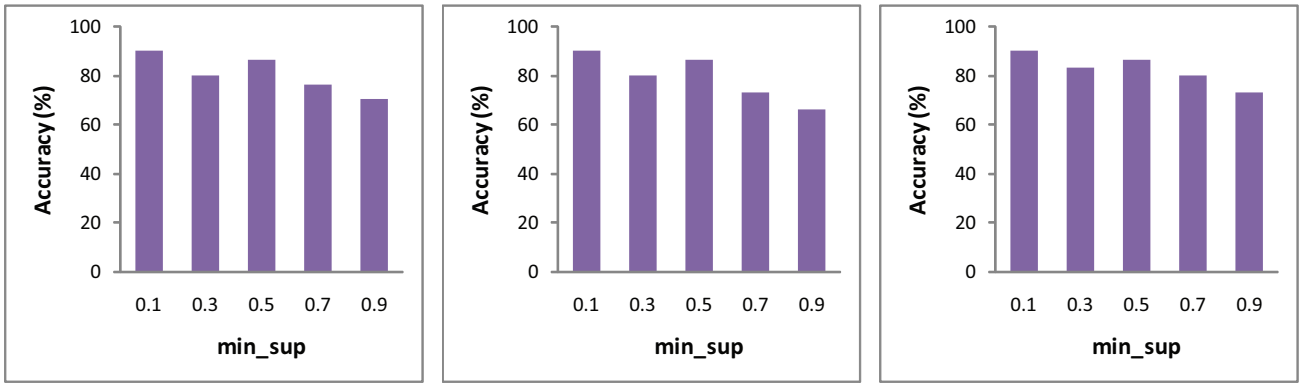
Figure 3: Accuracy vs. # of Intervals ($n = 6$, $m = 20$)

Figure 2 illustrates that the accuracy spans from 70% to 90% for 2 intervals, from 66% to 90% for 4 intervals, and from 73% to 90% for 6 intervals. Though we are testing a broad range of min_sup, the accuracy is relatively stable. These results suggest that our method is very robust to different user-specified min_sup. In the effort to study the effect of how the number of discretized intervals could on the accuracy, we measure the authorship identification accuracy with respect to the number of intervals. Figure 3 also suggests that our method is very robust to different number of intervals.

Figure 4 depicts the authorship identification accuracy for $n = 10$ and $m = 10$ (i.e., a total of 100 e-mails) on different number of discretized intervals. The accuracy spans from 80% to 90% at $min\_sup = 0.1$ and 0.3, suggesting that our proposed method can effectively identify the author of an e-mail based on the extracted write-prints when a reasonable min_sup is specified. As min_sup increases, the accuracy decreases as explained before.

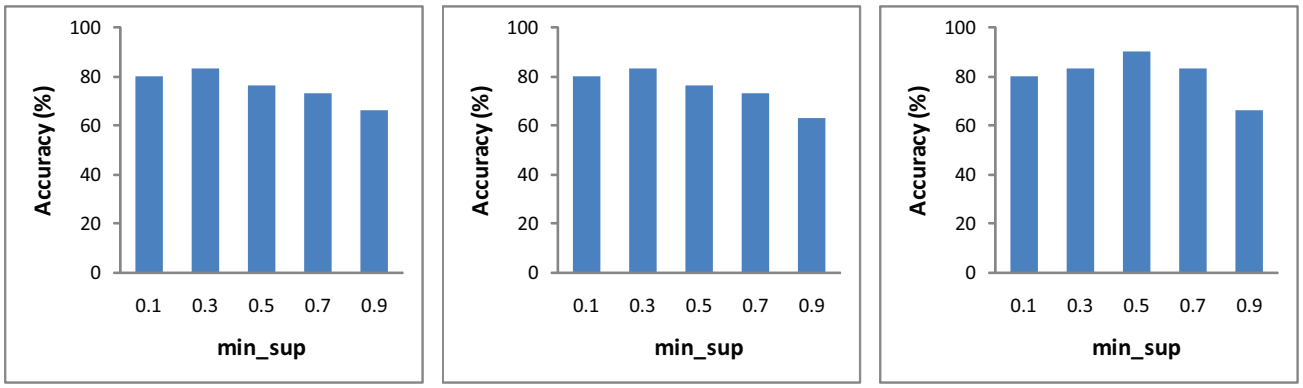
Figure 4 illustrates that the accuracy spans from 66% to 83% for 2 intervals, from 63% to 83% for 4 intervals, and from 66% to 90% for 6 intervals. Though we are testing a broad range of min_sup, the accuracy is relatively stable. These results suggest that our method is very robust to different user-specified min_sup. In the effort to study the effect of how the number of discretized intervals could on the accuracy, we measure the authorship identification accuracy with respect to the number of intervals. Figure 5 also suggests that our method is very robust to different number of intervals.

Comparing Figures 2 and 4, we notice that the authorship identification accuracy drops from the average



(a) # of Discretized Intervals = 2 (b) # of Discretized Intervals = 4 (c) # of Discretized Intervals = 6

Figure 2: Accuracy vs. *min_sup* **($n = 6$, $m = 20$)**



(a) # of Discretized Intervals = 2 (b) # of Discretized Intervals = 4 (c) # of Discretized Intervals = 6

Figure 4: Accuracy vs. *min_sup* **($n = 10$, $m = 10$)**

of 80.5% in Figures 2 to the average of 77% in Figures 4. Though there is a drop in accuracy, the drop is relatively small compared to the increase of suspects from 6 to 10. Most of traditional classifiers would have a very significant drop as the number of target classes (suspects) increases.

In additional to measuring the quality of write-print using authorship identification accuracy, we also manually examined the extracted write-print and found that frequent patterns can succinctly capture combinations of features that occur frequently in the suspect's emails. 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) e-mail, the frequent patterns themselves serve as a strong evidence for supporting the conclusion on authorship.

6. CONCLUSION

In this paper, we formally define the problem of authorship attribution and refine the problem into three subproblems: (1) To identify the write-print of each suspect. (2) To determine the author of the malicious e-mail. (3) To extract evidence for supporting the conclusion on authorship. Generally, the same three phased methodology is applied in the court of law for resolving the attribution issue. Most previous contributions focused on improving the classification accuracy of authorship identification, but only very few of them study how to gather strong evidence for the court of law.

We introduce a novel approach of authorship attribution and formulate a new notion of write-print based on the concept of frequent patterns. Unlike the write-prints in previous literature that are a set of predefined features, our notion of write-print is dynamically extracted from the data as combinations of features that occur frequently in a suspect's emails, but not frequently in other suspect's emails. The experimental results on real-life e-mail dataset suggest that the identified write-print does not only help identify the author of an anonymous e-mail, but also presents intuitive yet strong evidence for supporting the authorship finding.

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

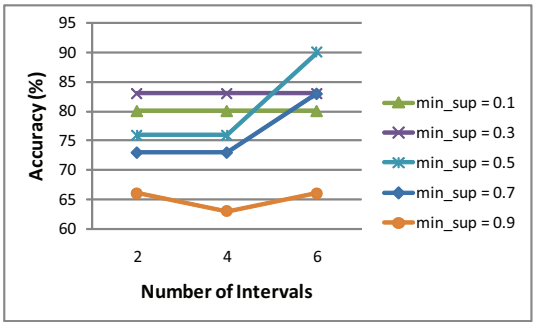


Figure 5: Accuracy vs. # of Intervals ($n = 10$, $m = 10$)

and utilize the concept of frequent pattern to identify hidden write-print of individuals for the purpose of e-mail forensics. Similarly, more interesting results can be obtained by using the proposed approach on real e-mail traffic containing malicious emails.

7. ACKNOWLEDGMENTS

The research is supported in part by the Discovery Grants (356065-2008) from the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Faculty Start-up Funds from Concordia University.

8. REFERENCES

[1] 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), March 2008.

[2] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, 22(2):207–216, June 1993.

[3] J. Burrows. An ocean where each kind...: Statistical analysis and some major determinants of literary style. computers and the. *Computers and the Humanities*, 23(4-5):309–321, August 1989.

[4] M. Corney, de Vel O., A. Anderson, and G. Mohay. Gender-preferential text mining of e-mail discourse. In *18th annual Computer Security Applications Conference (ACSAC*, 2002.

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

[6] O. De Vel. Mining e-mail authorship. paper presented at the workshop on text mining. In *ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, 2000.

[7] 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.

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

[9] 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, May 2003.

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

[11] J. Han and J. Pei. Mining frequent patterns by pattern-growth: methodology and implications. *ACM SIGKDD Explorations Newsletter*, 2(2), 2000.

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

[13] D. I. Holmes and R. S. Forsyth. The federalist revisited: New directions in authorship attribution.

[14] 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.

[15] G. R. Ledger and T. V. N. Merriam. Shakespeare, fletcher, and the two noble kinsmen. *Literary and Linguistic Computing*, 9:235–248, 1994.

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

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

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

[19] 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.

[20] G. Yule. On sentence length as a statistical characteristic of style in prose. biometrika, 30, 363390. 1938.

[21] G. Yule. The statistical study of literary vocabulary. cambridge, uk: Cambridge university press. 1944.

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

[23] 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*.

[24] R. Zheng, Y. Qin, Z. Huang, and H. Chen. Authorship analysis in cybercrime investigation. In *Proc. of the 1st International Symposium on Intelligence and Security Informatics (ISI)*, 2003.