Digital Investigation 8 (2012) 147–160

Contents lists available at SciVerse ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

Mining criminal networks from unstructured text documents

Rabeah Al-Zaidy [a], Benjamin C.M. Fung [a,*], Amr M. Youssef [a], Francis Fortin [b]

[a] Concordia Institute for Information Systems Engineering, Concordia University, 1455 De Maisonneuve Blvd. West, CIISE (EV7.640), Montreal, Québec, Canada H3G 1M8
[b] Sûreté du Québec, Montreal, Québec, Canada

ARTICLE INFO

Article history:
Received 14 October 2010
Received in revised form 16 October 2011
Accepted 26 December 2011

Keywords:
Forensic analysis
Data mining
Hypothesis generation
Criminal network
Information retrieval

ABSTRACT

Digital data collected for forensics analysis often contain valuable information about the suspects' social networks. However, most collected records are in the form of unstructured textual data, such as e-mails, chat messages, and text documents. An investigator often has to manually extract the useful information from the text and then enter the important pieces into a structured database for further investigation by using various criminal network analysis tools. Obviously, this information extraction process is tedious and error-prone. Moreover, the quality of the analysis varies by the experience and expertise of the investigator. In this paper, we propose a systematic method to discover criminal networks from a collection of text documents obtained from a suspect's machine, extract useful information for investigation, and then visualize the suspect's criminal network. Furthermore, we present a hypothesis generation approach to identify potential indirect relationships among the members in the identified networks. We evaluated the effectiveness and performance of the method on a real-life cybercrimine case and some other datasets. The proposed method, together with the implemented software tool, has received positive feedback from the digital forensics team of a law enforcement unit in Canada.

1. Introduction

In many criminal cases, computer devices owned by the suspect, such as desktops, notebooks, and smart phones, are target objects for forensic seizure. These devices may not only contain important evidences relevant to the case under investigation, but they may also have important information about the social networks of the suspect, by which other criminals may be identified. In the United States, the FBI Regional Computer Forensics Laboratory (RCFL) conducted over 6000 examinations on behalf of 689 law enforcement agencies across the United States in one year (RCFL, 2009). The amount of data they examined in 2009 has reached 2334 Tera Bytes (TB), which is a double of the size processed in 2007. To accommodate the increasing demand, better resources are needed to help investigators process forensically collected data.

Most collected digital evidence are often in the form of textual data, such as e-mails, chat logs, blogs, webpages, and text documents. Due to the unstructured nature of such textual data, investigators usually employ some off-the-shelf search tools to identify and extract useful information from the text, and then manually enter the useful pieces into a well-structured database for further investigation. Obviously, this manual process is tedious and error-prone; the completeness of a search and the quality of an analysis pretty much relies on the experience and expertise of the investigators. Important information may be missed if a criminal intends to hide it.

In this paper, we propose a data mining method to discover criminal communities and extract useful information for investigation from a collection of text documents obtained from a suspect's machine. The objective is

* Corresponding author. Tel.: +1 514 8482424.
E-mail address: fung@ciise.concordia.ca (B.C.M. Fung).

1742-2876/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.diin.2011.12.001

to help investigators efficiently identify relevant information from a large volume of unstructured textual data. The method is especially useful in the early stage of an investigation when investigators may have little clue to begin with. The effectiveness of the proposed method, together with the implemented software tool, have received positive feedback from the digital forensics team of a law enforcement unit in Quebec, Canada. Our major contributions can be summarized as follows.

1. *Communities discovery from unstructured textual data*. Several social network analysis tools (Getoor and Diehl, 2005; Xu and Chen, 2005) are available to assist investigators in the analysis of criminal networks. However, these tools often assume that the input is a structured database. Nonetheless, structured data is often not available in real-life investigations. Instead, the available input is usually a collection of unstructured textual data. Our first contribution is to provide an end-to-end solution to automatically discover, analyze, and visualize criminal communities from unstructured textual data.
2. *Introduction of the notion of prominent communities*. After extensive discussions with the digital forensics team of a Canadian law enforcement unit, we defined the notions of *community* and *prominent community*. In the context of this paper, two or more persons form a *community* if their names appear together in at least one investigated document. A community is *prominent* if its associated names frequently appear together in some minimum number of documents, which is a user-specified threshold. We propose a method to discover *all* prominent communities and measure the closeness among the members in these communities.
3. *Generation of indirect relationship hypotheses*. The notions of prominent community and closeness among its members capture the *direct relationships* among the persons identified in the investigated documents. Our recent work (Al-Zaidy et al., 2011) presents a preliminary study on direct relationships. In many cases, indirect relationships are also interesting since they may reveal hidden relationships. For example, person A and person B are indirectly related if both of them have mentioned a meeting at hotel X in their written e-mails, even though they may not have any direct communications. We present a method to generate all indirect relationship hypotheses with a maximum, user-specified, depth.
4. *Scalable computation*. The computations of prominent communities and closeness from the investigated text document set is non-trivial. A naive approach is to enumerate all $2^{|U|}$ combinations of communities and scan the document set to determine the prominent communities and the closeness, where $|U|$ is the number of distinct personal names identified in the input document set. Our proposed method achieves scalable computation by efficiently pruning the non-prominent communities and examining the closeness of the ones that can potentially be prominent. The scalability of our method is supported by experimental results.

The rest of the paper is organized as follows. Section 2 reviews the related works. The problems of criminal community discovery and indirect relationship hypotheses generation are formally defined in Section 3 and our proposed method is described in Section 4. Section 5 demonstrates the effectiveness of our proposed method via a case study on real-life cybercrime investigation. Section 6 shows the performance study of our proposed method. Section 7 concludes the paper.

2. Related works

Criminal network analysis has received great attention from researchers. The pioneer work by Chen et al. (2004) demonstrates a successful application of data mining techniques to extract criminal relations from a large volume of police department's incident summaries. They use the co-occurrence frequency to determine the weight of relationships between *pairs* of criminals. Yang and Ng (2007) present a method to extract criminal networks from web sites that provide blogging services by using a topic-specific exploration mechanism. In their approach, they identify the actors in the network by using web crawlers that search for blog subscribers who participated in a discussion related to some criminal topics. After the network is constructed, they use some text classification techniques to analyze the content of the documents. Finally they propose a visualization of the network that allows for either a concept network view or a social network view. Our work is different from these works in three aspects. First, our study focuses on unstructured textual data obtained from a suspect's hard drive, not from a well-structured police database. Second, our method can discover prominent communities consisting of any size, i.e., not limited to pairs of criminals. Third, while most of the previous works focus on identifying direct relationships, the methods presented in this paper can also identify indirect relationships.

A criminal network follows a social network paradigm. Thus, the approaches used for social network analysis can be adopted in the case of criminal networks. Many studies have introduced various approaches to construct a social network from text documents. Hope et al. (2006) propose a framework to extract social networks from text document that are available on the web. Jin et al. (2009) propose a method to rank companies based on the social networks extracted from webpages. These approaches rely mainly on web mining techniques to search for the actors in the social networks from web documents. Another direction of social network studies targets some specific type of text documents such as e-mails. Zhou et al. (2006) propose a probabilistic approach that not only identifies communities in email messages but also extracts the relationship information using semantics to label the relationships. However, the method is applicable to only e-mails and the actors in the network are limited to the authors and recipients of the e-mails.

Researchers in the field of knowledge discovery have proposed methods to analyze relationships between terms in text documents in a forensic context. Jin et al. (2007) introduce a concept association graph-based approach to

search for the best evidence trail across a set of documents that connects two given topics. Srinivasan (2004) proposes the open and closed discovery algorithms to extract evidence paths between two topics that occur in the document set but not necessarily in the same document. Skillicorn and Vats (2007) employ the open discovery approach to search for keywords provided by the user and return documents containing other different but related topics. They further apply clustering techniques to rank the results and present the user with clusters of new information that are conceptually related to their initial query terms. Their open discovery approach searches for novel links between concepts from the web with the goal of improving the results of web queries. In contrast, this paper focuses on extracting information for investigation from text files.

3. Problem description

The problem of criminal networks analysis can be divided into two problems. The first one is to discover the prominent communities in a document set and extract useful information from the documents that contribute to the formation of the prominent communities. The second one is to generate hypotheses of indirect relationships between the prominent communities and other people names in the document set. These two problems are formally defined as follows.

3.1. The problem of criminal community discovery

The problem of criminal community discovery is to identify the hidden communities from a collection of text documents obtained from one (or multiple) suspect's file systems. In this paper, a text document is generally defined to be a logical unit of textual data, such as an e-mail message, a chat session, a webpage, a blog session, and a text file. Let D be a set of input text documents. Let U be the set of distinct personal names identified in D. Each document $doc \in D$ is represented as a set of names such that $doc \subseteq U$ Let $C \subseteq U$ be a set of personal names called a *community*. A document *doc contains* a community C if $C \subseteq doc$ A community having k personal names is a *k-community*. The *support* of a community C is the number of documents in D containing C. For example, {*Alan*, *Kim*} in Table 1 is a 2-community with support = 1. A community C is a *prominent community* in a set of documents D if the support of C is greater than or equal to a user-specified minimum support threshold. Suppose the threshold is set to 2. Then, {*Alan*, *Kim*} is not a prominent community in Table 1, but {*Jenny*, *John*, *Mike*} is a prominent 3-community with support = 2.

Table 1
Document set (D).

Document	Names in doc_i
doc_1	{*Alan*, *John*, *Kim*}
doc_2	{*Jenny*, *John*, *Mike*}
doc_3	{*Alan*, *Jenny*, *John*, *Mike*}
doc_4	{*Jenny*, *Mike*}

Definition 3.1 (Prominent community). Let D be a set of text documents. Let $support(C)$ be the number of documents in D that contain C, where $C \subseteq U$ A community C is a *prominent community* in D if $support(C) \geq min\_sup$, where the minimum support min_sup is a user-specified positive integer threshold.

The identified prominent communities are also called the *criminal communities* in this paper because the document set is assumed to be obtained from the suspect's file system under investigation. The problem of criminal community discovery is formally defined as follows:

Definition 3.2 (Problem of criminal community discovery). Let D be a set of text documents. Let min_sup be a user-specified minimum support threshold. The *problem of criminal community discovery* is to identify all prominent communities from D with respect to min_sup, and to extract useful information from the documents of every prominent community for crime investigation.

The specific type of information that is useful for investigation depends on the specific criminal case in hand. We will elaborate this point in Section 4.

3.2. The problem of indirect relationship hypothesis generation

A person is indirectly related to a prominent community if there exists a sequence of intermediate terms that links a person and a prominent community through a chain of documents, in which the starting document and the ending documents contain the prominent community and the personal name, respectively. The problem of indirect relationship hypothesis generation is to identify all indirect relationships. Specifically, an indirect relationship consists of a sequence of intermediate terms between a prominent community and a personal name identified in the given document set. The generated indirect relationships may reveal some hidden links that the investigator might not be aware of. Yet, they are only hypotheses; the investigator has to further verify the truthfulness and usefulness of these relationships.

Definition 3.3 (Indirect relationship). Let D be a set of documents. Let U be a set of distinct names identified in D. Let $C \subseteq U$ be a prominent community and $p \in (U - C)$ be a person name that is not in C. Let $D(\cdot) \subseteq D$ denote the set of documents containing the enclosed argument where the enclosed argument can be a community, a personal name, or a text term. Let $D(C)$ and $D(p)$ be the sets of documents in D that contain C and p, respectively. An indirect relationship of depth d between C and p is defined by a sequence of terms $[t_1, \ldots, t_d]$ such that

1. $D(C) \cap D(p) = \emptyset$
2. $(t_1 \in D(C)) \wedge (t_d \in D(p))$
3. $(t_r \in D(t_{r-1})) \wedge (t_r \in D(t_{r+1}))$ for $1 < r < d$
4. $D(t_{r-1}) \cap D(t_{r+1}) = \emptyset$ for $1 < r < d$

Condition (1) requires that a prominent community C and a personal name p do not co-occur in any document. Condition (2) states that the first term t_1 must occur in at

least one document containing C and the last term t_d must occur in at least one document containing p. Condition (3) requires that the intermediate terms t_r must co-occur with the previous term t_{r-1} in at least one document, and t_r must co-occur with the next term t_{r+1} in at least one document. This requirement defines the chain of documents linking C and p. Condition (4) requires that the previous term t_{r-1} and the next term t_{r+1} do not co-occur in any document. The problem of indirect relationship hypothesis generation is formally defined as follows:

Definition 3.4 (Problem of indirect relationship hypothesis generation). Let D be a set of text documents. Let U be the set of distinct personal names identified in D. Let G be the set of prominent communities discovered in D according to Definition 3.2. The *problem of indirect relationship hypothesis generation* is to identify all indirect relationships of maximum depth *max_depth* between any prominent community $C \in G$ and any personal name $p \in U$ in D, where *max_depth* is a user-specified positive integer threshold.

4. Our method

Fig. 1 depicts an overview of our proposed *Criminal Community Mining System* (*CCMS*). The first step is to read the investigated text documents and extract the personal names from them. The name extraction task is followed by a normalization process to eliminate duplicate names that refer to the same person. The next step is to discover the prominent criminal communities from the extracted names. Then, we extract the profile information that is valuable to investigators, such as contact information and summary topics, from the documents that contribute to the prominent communities. Next, we search for indirect relationships between the criminals across the document set. Finally, we provide a visual representation of the prominent communities, their related information, and the indirect relationships found in the document set. Below, we elaborate the steps of prominent community discovery, community information extraction, and indirect relationship generation.

4.1. Identifying prominent communities

The first step is to identify the personal names from the input document files. There are many Named Entity Recognition (NER) tools and methods available in the market to extract personal names. In our system, we adopt the Stanford Named Entity Tagger (Finkel et al., 2005), which is a promising tool for identifying English names. For each document $doc \in D$, we apply the NER tagger to obtain a set of personal names in *doc*. Variant names of the same

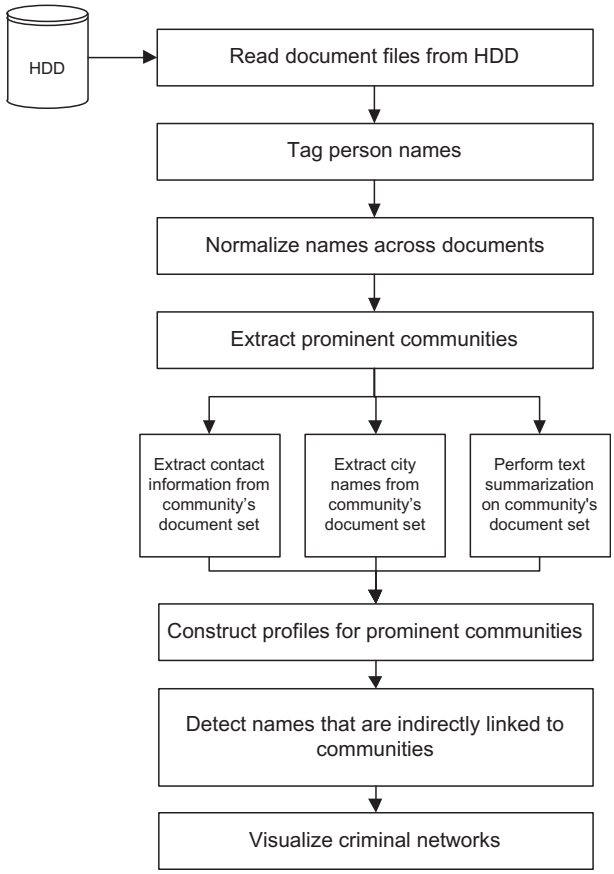

Fig. 1. Criminal community mining system.

person are merged into one name. For instance, *John, J. Smith*, and *John Smith* are transformed into a common form *John Smith*. Our method also allows the user to incorporate his/her domain knowledge to merge the names. For instance, some people use nicknames in a chat log and their real name is mentioned in the same session. Other NER tools can be employed if the document files contain non-English names; however, NER is not the focus of this paper.

The next step is to identify the prominent criminal communities. When two or more persons interact frequently or their names appear together frequently, this indicates a strong direct linkage. Analyzing the strength of linkages is a key step for effective crime investigation. The strength of a linkage can be measured by comparing the frequency of the interaction between the individuals to a fixed threshold. A linkage is strong if the number of interactions passes a given threshold; otherwise, the linkage is weak or there is no linkage. A community is considered to be a prominent community if its support is equal to or greater than a given threshold.

A naive approach to identify all prominent communities is to enumerate all possible communities and identify the prominent ones by counting the support of each community in D. Yet, in case the number of identified personal names $|U|$ is large, it is infeasible to enumerate all possible communities because there are $2^{|U|}$ possible combinations. To efficiently extract all prominent communities from the set of identified individuals, we modify the Apriori algorithm (Agrawal et al., 1993), which is originally designed to extract frequent patterns from transaction data.

Recall that U denotes the universe of all personal names in D, and each document $doc \in D$ is represented as a set of names such that $doc \subseteq U$. Our proposed algorithm, *Prominent Community Discovery* (*PCD*), is a level-wise iterative search algorithm that uses the prominent k-communities to explore the prominent $(k+1)$-communities. The generation of prominent $(k+1)$-communities from prominent k-communities is based on the following PCD property.

Property 4.1 (PCD property). All nonempty subsets of a prominent community must also be prominent because $support(C') \geq support(C)$ if $C' \subseteq C$.

By definition, a community C' is not prominent if $support(C') < min\_sup$. The above property implies that adding a personal name p to a non-prominent community C' will never make it prominent. Thus, if a k-community C' is not prominent, then there is no need to generate $(k+1)$-community $C' \cup \{p\}$ because $C' \cup \{p\}$ must not be prominent. The strength of the linkages among the members in a prominent community C is indicated by $support(C)$. The presented algorithm can identify *all* prominent communities by efficiently pruning the communities that cannot be prominent based on the PCD property.

Algorithm 1. Prominent Community Discovery

Input: A set of text documents D.
Input: User-specified minimum support min_sup.
Output: Prominent communities $G = \{G_1, \ldots, G_k\}$.
Output: $support(C_j)$ and $D(C_j)$ for every $C_j \in G$.

```
1:  G_1 = all prominent 1-communities in D;
2:  for (k = 2; G_{k-1} ≠ ∅; k++) do
3:    Candidates_k = G_{k-1} ⋈ G_{k-1};
4:    for all community C ∈ Candidates_k do
5:      if ∃C' ⊂ C such that C' ∉ G_{k-1} then
6:        Candidates_k = Candidates_k − C;
7:      end if
8:    end for
9:    support(C_j) = 0 and D(C_j) = ∅ for every C_j ∈ Candidates_k;
10:   for all document doc_i ∈ D do
11:     for all C_j ∈ Candidates_k do
12:       if C_j ⊆ doc_i then
13:         support(C_j) = support(C_j) + 1;
14:         D(C_j) ← doc_i;
15:       end if
16:     end for
17:   end for
18:   G_k = {C_j ∈ Candidates_k | support(C_j) ≥ min_sup};
19: end for
20: return G = {G_1, ..., G_k} with support(C_j) and D(C_j);
```

Algorithm 1 summarizes our Prominent Community Discovery algorithm. The algorithm finds the prominent k-communities from the prominent $(k-1)$-communities based on the PCD property. The first step is to find the set of prominent 1-communities, denoted by G_1. This is achieved by scanning the document set once and counting the support count for each 1-community C_j. G_1 contains all prominent 1-communities C_j with $support(C_j) \geq min\_sup$. The set of prominent 1-communities is then used to identify the set of candidate 2-communities, denoted by $Candidates_2$. Then the algorithm scans the database once to count the support of each candidate C_j in $Candidates_2$. All candidates C_j that satisfy $support(C_j) \geq min\_sup$ are prominent 2-communities, denoted by G_2. The algorithm repeats the process of generating G_k from G_{k-1} and stops if $Candidate_k$ is empty.

Personal names in a community are sorted by lexicographical order. Two prominent $(k-1)$-communities can be joined together to form a candidate k-community only if their first $(k-2)$ personal names are identical and their last $(k-1)$ personal names are different. This operation is based on the PCD property: A community cannot be prominent if any of its subsets is not prominent. Thus, the only potential prominent communities of size k are those that are formulated by joining prominent $(k-1)$-communities. Lines 4–8 describe the procedure of removing candidates that contain at least one non-prominent $(k-1)$-community.

Lines 9–17 describe the procedure of scanning the database and obtaining the support count of each community C_j in $Candidates_k$. Each candidate community C_j is looked up in each document doc_i in the document set. Once a match is found, the value of $support(C_j)$ is incremented by 1 and the document doc_i is added to the set $D(C_j)$. If $support(C_j)$ is greater than or equal to the user-specified minimum threshold min_sup, then C_j is added to G_k, the set of prominent k-communities with k members. The algorithm terminates when no more candidates can be generated or when none of the candidate communities pass the min_sup threshold. The algorithm returns all prominent communities $G = \{G_1, \ldots, G_k\}$ with support counts and sets of associated documents for each prominent community.

Example 4.1 (Prominent communities discovery). Consider Table 1 with $min\_sup = 2$. First, we scan the table to find $G_1 = \{Alan, Jenny, John, Mike\}$. Next, we perform $G_1 \bowtie G_1$ to generate $Candidates_2 = \{\{Alan, Jenny\}, \{Alan, John\}, \{Alan, Mike\}, \{Jenny, John\}, \{Jenny, Mike\}, \{John, Mike\}\}$. Then we scan the table once to obtain the support of every community in $Candidates_2$ with $support \geq 2$, and identify $G_2 = \{\{Alan, John\}, \{Jenny, John\}, \{Jenny, Mike\}, \{John, Mike\}\}$. Similarly, we perform $G_2 \bowtie G_2$ to generate $Candidates_3 = \{\{Jenny, John, Mike\}\}$ and scan the table once to identify the prominent 3-community $G_3 = \{\{Jenny, John, Mike\}\}$.

4.2. Extracting information of prominent communities

The next phase is to retrieve useful information for crime investigation, such as contact information, from the discovered prominent communities. In the context of this paper, a group of people are considered to be in the same prominent community if their names appear together frequently in a minimum number of text documents. Thus, the topics of the set of documents containing their names are the "reasons" bringing them together. By analyzing the content of the text documents containing the names of the community members, a crime investigator may obtain valuable clues that are useful for further investigation, especially during the early stages of the investigation. For instance, if a set of community member names are all contained in the same chat sessions, then summarizing the topics of the discussion can help the investigator infer the type of relationship the community members share. To facilitate the crime investigation process, we extract the following types of information from the set of documents $D(C_j)$ for each prominent community C_j:

1. Key topics
2. Names of other people who are not in C_j
3. Locations and addresses
4. Phone numbers
5. E-mail addresses
6. Website URLs.

In some real-life cyber criminal cases, there could be thousands of identified individuals and hundreds of prominent communities. Even with a data mining software, an investigator may still find it difficult to cope with such a large volume of information. The summarized key topics from $D(C_j)$ can provide the investigator with an overview of each community and the related topics. The extracted key topics can be a link label when the communities are visualized on the screen. Some people names may appear only a few times in $D(C_j)$ but may not be frequent enough to be included as a member in C_j. Identifying these infrequent people names may lead to some new clues for investigation. Locations, addresses, and contact information, such as phone numbers and e-mail addresses, are valuable information for crime investigation because they may reveal other potential channels of communications among members of the criminal community.

To extract the key topics, we employ an Open Text Summarizer (OTS) (Rotem, 2003). To extract the city names, we search the documents for the cities in the GeoWorldMap database (Geobytes Inc., 2003). To extract other addresses, phone number, and e-mail addresses, we use regular expressions (Friedl, 2006).

Other useful information may be extracted to further describe the relationship between the members of an identified prominent community, such as the *duration* of the relationship which is a key piece of information regarding the activity of members of a community. It is especially useful to provide the investigator with a sense of a time line for the relationships that the communities share. To specify the duration of the relationship between a criminal community identified in a set of text documents, we make use of the metadata of the documents. The metadata of a file is the data linked to this

file by the hosting system upon creation of the document. We can define the duration of a relationship as all or some of the values of: (1) the starting date of the relationship, (2) the ending date, (3) and the amount of time the relationship lasted. We can identify the starting date of the relationship between members of a prominent community C_j, by the oldest of the dates attached to the documents in $D(C_j)$. The end date of the relationship is the most recent of the dates associated with the documents. The duration of the relationship is then calculated as the difference between the start and end dates.

4.3. Discovering indirect relationships

In this section, we present a method to discover the evidential trails between a prominent community identified in a dataset and other people in the document set who are not in the community. An evidential trail represents a relationship between the prominent community and other people through a common topic rather than co-occurrence. This trail is extracted as a chain of intermediate terms that link a community to a person. Thus, for a given prominent community C_j and a personal name p, the indirect relationship discovery method identifies a chain of intermediate terms t from the dataset that links C_j with p. The length of the chain is limited by a user-specified threshold, denoted by *max_depth*.

4.3.1. Profiles

Any term t in a document set D can be profiled by extracting interesting information about it from the textual content of documents in D. For example, if the document set is obtained from newswire documents, the profile of a topic such as *Microsoft* can be: *Corporation*, *Windows*, *Bill Gates*, and *Office*. In the same sense, the profile for a prominent community C existing in a hard drive can be *city*, *phone*, and *email*. This information can be retrieved from the documents in which the prominent community occurs. However, this information should not be chosen randomly because of the importance of the profile information in the hypothesis discovery process. If the profile information is too general, the discovered relationship is unlikely to be significant and the investigator may be overwhelmed by a large number of false hypotheses. Thus, data must only be added to the profile if it satisfies some pre-specified constraints or conditions that are set to ensure the usefulness of this data.

The structure of a profile is based on semantic types. This structure ensures that only a specific type of information is added to the profile. In the criminal network analysis context, we select semantic types that are significant to investigations. In particular, the following semantic types are selected: (1) summary topics of the documents representing the prominent community's interactions, (2) other names of people mentioned in the documents with the prominent communities, (3) cities and locations, (4) email addresses, (5) phone numbers, and (6) website URLs. These semantic types are also used to identify the relationship between the members of prominent communities. For the profiles of the prominent communities, we use the same information that is retrieved for the prominent communities, as explained earlier, for several reasons. First, it is less costly in the information extraction process. Second, these semantic types are extracted as forensically valuable information about the set of related individuals and consequently any other relationships that are found using this information are likely to be valuable as well. Within each semantic type in the profile of a prominent community, each term has a weight associated with it. In order to minimize the computations, we define the profiles for the prominent communities of maximal size, and combine all the profiles of the sub-communities.

Definition 4.1 (Profile). A profile for a prominent community C, denoted by $P(C)$, is defined by a collection of vectors $V_{x_1}, V_{x_2}, \ldots, V_{x_n}$, where n denotes the number of semantic types considered. Each vector V_{x_i}, of length l_{x_i}, where l_{x_i} is the number of terms of semantic type x_i, is given by:

$$V_{x_i} = \begin{bmatrix} t_1, f_{x_i}(t_1) \\ t_2, f_{x_i}(t_2) \\ \vdots \\ t_{l_{x_i}}, f_{x_i}\left(t_{l_{x_i}}\right) \end{bmatrix}$$

where $f_{x_i}(t_j)$ is the weight of term t_j and is given by

$$f_{x_i}(t_j) = \frac{f'_{x_i}(t_j)}{\max_j f'_{x_i}(t_j)}$$

and

$$f'_{x_i}(t_j) = n_{x_i,t_j} \times \log\left(|D|/n_{t_j}\right),$$

where $|D|$ is the total number of documents, n_{t_j} is the frequency of occurrence for term t_j in the document set D, and n_{x_i,t_j} is the frequency of term t_j of semantic type x_i in $D(C)$.

4.3.2. Indirect relationship generation algorithm

Given a prominent community C with profile $P(C)$, we propose the indirect relationships generation algorithm to extract indirect relationships between the prominent community C and other individuals in the document set. This algorithm is a hybrid version of both the open and closed discovery algorithms described in Srinivasan (2004). The closed discovery requires two terms, A and C, and generates hypothetical relationships between A and C through intermediate terms B. On the other hand, open discovery requires the entry of only one initial term; the B and C are provided by the algorithm. We propose a model that is open in the sense that it requires only one initial term to start the discovery process. However, the other end of the relationship must be the name of an individual from the document set.

Algorithm 2. Indirect Relationship Generation Algorithm

Input: Profile $P(C) = \{V_{x_1}, \ldots, V_{x_n}\}$ for community C, number of intermediate terms N, maximum depth threshold max_depth
Output: Set of personal names indirectly related to C through intermediate terms

```
P = P(C);
dep = max_depth;
while dep > 0 do
    Let B_{x_i} denote the N top ranking terms in V_{x_i} corresponding to P;
    Construct profile P(B_x[y]), x ∈ X, y = 1, ..., N;
    Combine profiles P(B_x[y]) into one profile P(O), where the weight of a term
    in P(O) is the sum of its weights in profiles P(B_x[y]), y = 1, ..., N;
    for each t_j ∈ V_x of P(O), x ∈ X do
        Conduct a query (t_j AND C) in D;
        if result ≠ ∅ then
            remove t_j from P(O);
        end if
    end for
    dep = dep − 1;
    P = P(O);
end while
return terms O_x ranked by weight, for semantic type x=persons;
```

Algorithm 2 shows the steps of the indirect relationship generation algorithm. The method is applied for each prominent community $C_j \in G$. The algorithm requires the profile of the prominent community $P(C)$ as input, where at least one of its term vectors V_x is of the semantic type persons. Both the number of intermediate terms N and the depth of the indirect relationship *max_depth* are set by the user. If the depth is 1, for example, then the indirect relationship between community a and person c is through one connecting term, e.g., $a \rightarrow b \rightarrow c$. However, if the depth is 2, then the relationship is of the form $a \rightarrow b \rightarrow e \rightarrow c$.

The algorithm proceeds with the truncation of the term vectors, V_x, comprising the profile $P(C)$, by selecting the top N ranking terms in each V_x for all values of $x \in X$. The new truncated vectors are called B_x for each semantic type x accordingly. Next, for each $x_i \in X$, we search the document set for each term $B_x[y]$, $y = 1,\ldots, N$ in order to build its profile. The constructed profiles are $P(B_x[1])$, $P(B_x[2])$, …, $P(B_x[N])$. Now, a combined profile is computed where the combined weight of a term is the sum of its weights in each of $P(B_x[1])$, $P(B_x[2])$, …, $P(B_x[N])$ in which it occurs. This combined profile is called $P(O)$ and is comprised of vectors V_x for each semantic type $x \in X$. For each term in t_j in the profile $P(O)$, if a search for (C AND t_j) returns a nonempty set, the term t_j is removed from $P(O)$. If the depth is set to a value greater than 1, the algorithm iterates again using the value of the profile $P(O)$ produced from the previous iteration as the input profile for the next iteration. Finally, the method returns the terms in $P(O)$ for the semantic type persons ranked by combined weight and terminates.

Example 4.2 (Indirect relationship hypothesis generation). To illustrate the steps of the algorithm, consider the community C with profile $P(C)$ in Fig. 2 with $depth = 1$. First, start with the profile $P(C)$ for the community $C = \{John, Jenny, Kim\}$ and construct profiles for each term in $P(C)$. The profiles for the terms *auction* and *seattle* are denoted by $P(auction)$ and $P(seattle)$, respectively, with values as shown in Fig. 3. Next, the profiles are combined into one profile $P(O)$ as shown in Fig. 3. For each name t_j in the persons vector, search for documents containing both t_j and C. In this example, the first lookup searches for documents containing all four names: *John*, *Jenny*, *Kim*, and *Sam*. If no document contains all of them, then it implies that *Sam* has no co-occurrence relationship with the prominent community. Thus, *Sam* is indirectly linked to C through the term auction and *Bob* is linked to C through the term *seattle*. Fig. 3 shows the final results of the discovery method when applied to this example.

5. Case study on real-life cybercrime

The objective of this case study is to demonstrate the effectiveness of our proposed notions and methods in

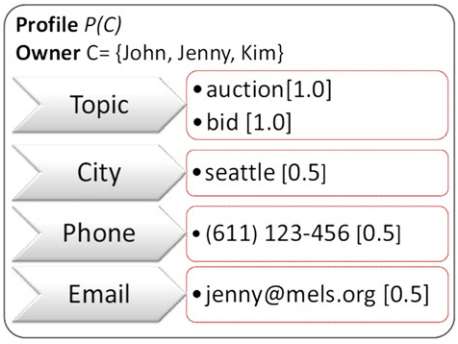

Fig. 2. Example of a profile.

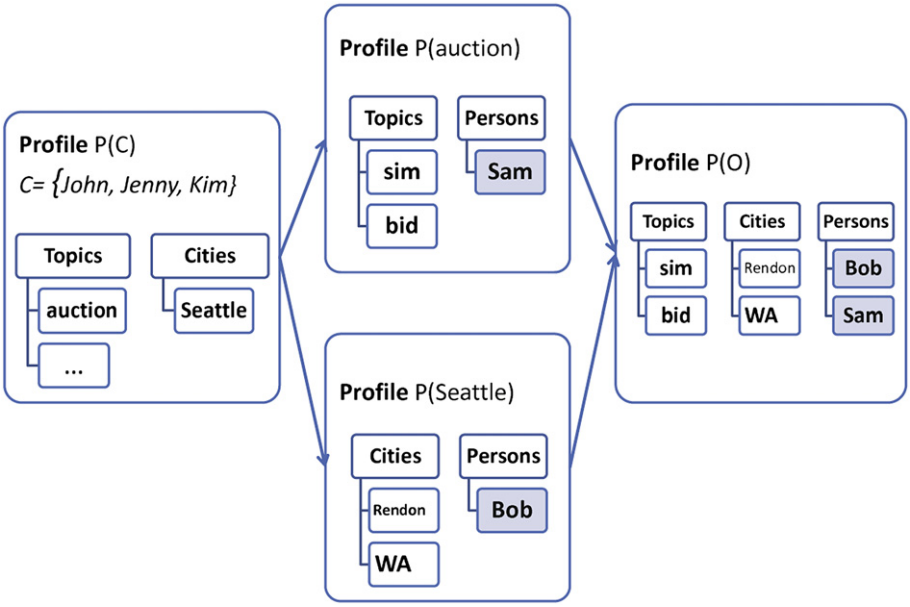

Fig. 3. Indirect relationship hypothesis generation.

a real-life cybercrime investigation. The dataset was provided by a Canadian law enforcement unit. In particular, we performed experiments on an MSN chat log from a hard disk that was confiscated from a suspect in a computer hacking case. The chat log, with a size of approximately 500 MB, contains the chat messages and file attachments from 220 distinct chat accounts. The case had already been solved by the investigator of the law enforcement unit. To judge the effectiveness of our proposed method, we compared the investigation result using our implemented prototype with the result manually obtained by the investigator. We were informed that the nature of the crime was related to computer hacking. No other information was provided to us regarding the chat log to be analyzed. This scenario is similar to the early stage of an investigation when an investigator has limited prior knowledge about the suspect(s). Due to confidentiality and privacy concerns, some of the information had to be masked and all the identities, e-mail accounts, and server names had to be replaced by pseudonyms in the following discussion.

Fig. 4 depicts the prominent communities identified from the MSN chat log with $min\_sup = 5$. Each node in the figure represents a distinct chat account. The distances among the nodes in a community C represent the closeness of its members, which are computed from the inverse of $support(C)$. Specifically, there are 24 distinct chat accounts, forming 32 prominent 2-communities, 4 prominent 3-communities, and 1 prominent 4-community. The

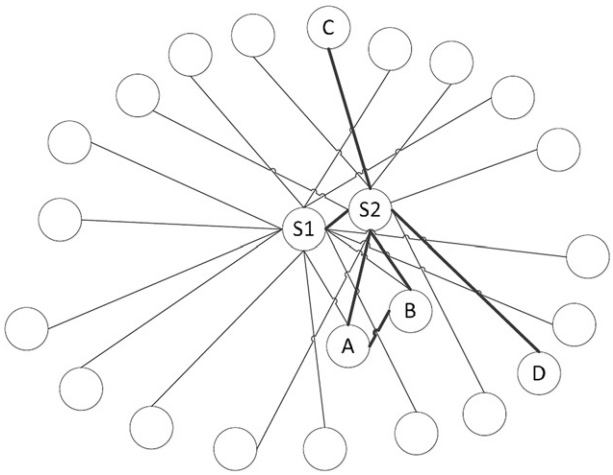

Fig. 4. Prominent communities in the case study.

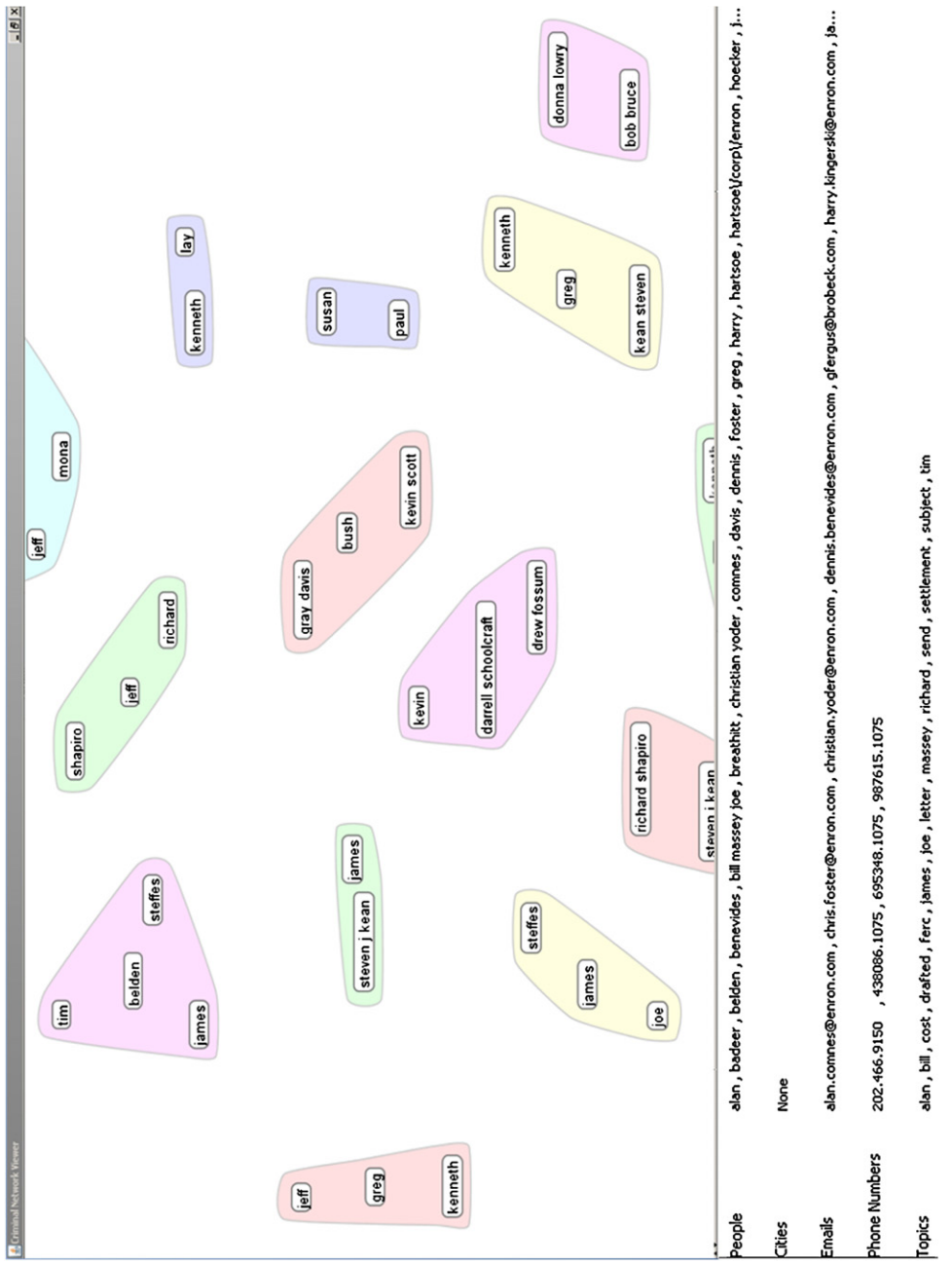

Fig. 5. Prominent communities in *EnronSmall*.

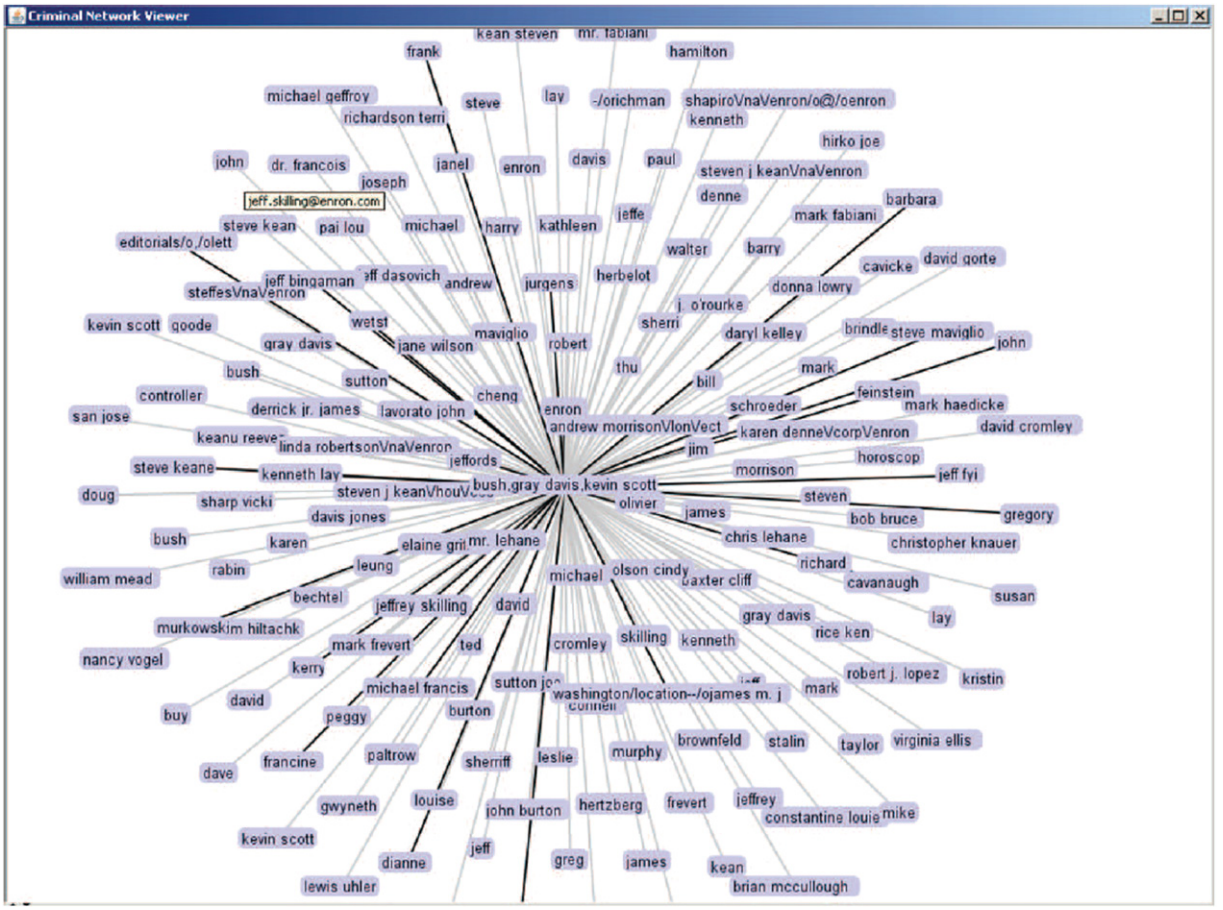

Fig. 6. Indirect relationships in *EnronSmall*.

interactions of the other 196 accounts are not frequent enough to form a prominent community. The two central nodes, denoted by *S*1 and *S*2, represent two chat log accounts owned by the same suspect, the owner of the confiscated computer; therefore, the other 22 users communicate with at least one of *S*1 and *S*2.

Recall from Section 4.2 that our proposed framework extracts some information, namely key topics, person names, locations, addresses, e-mails, and URLs, from the documents (the chat messages) of each prominent community. By performing a simple search on the key topics of each community, we identified a suspicious user,

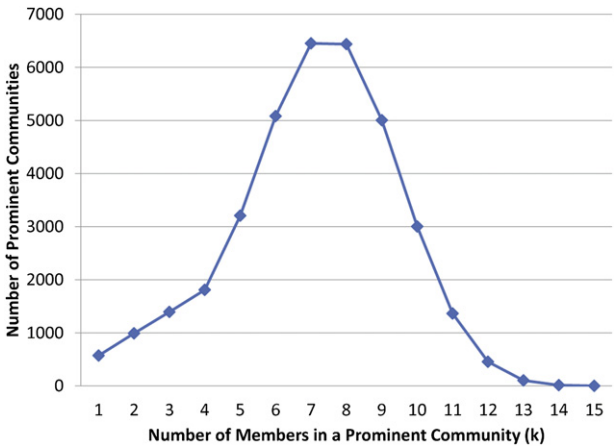

Fig. 7. Number of prominent communities vs. *k* in *EnronFull*.

denoted by *C*, who discussed about "botnet" with *S*2. This led us to further look into the details of the extracted community information of this particular prominent 2-community. We found that *S*2 and *C* had exchanged several e-mail addresses in a form similar to this anonymous form "aaa999@123.456.789.101.dsl.isp.ca". The subdomain and domain of the e-mail address indicate that it is a temporary IP address assigned by a DHCP server. Yet, it is unusual to have an e-mail server running on a dynamically changing IP address. Therefore, we alleged that the servers were not real e-mail servers, but some bots controlled by the suspect. Furthermore, *A* and *C* exchanged several suspicious URLs, in a form similar to "http:// <user>. <free hosting company>.com/save.exe", which point to some binary executable files. We concluded that these executables were probably used for spreading the malware to victims. The indirect relationships discovery process also illustrates that *C* is indirectly related to the prominent 4-community {*S*1, *S*2, *A*, *B*} through the shared URLs.

Among the 37 prominent communities, we identified 6 prominent 2-communities and 1 prominent 3-community that share suspicious information similar to the aforementioned e-mail addresses and URLs. These suspicious communications are indicated by the dark lines in Fig. 4. We confirmed the correctness of these identified criminal communities and activities with two cybercrime investigators in the law enforcement unit who solved the real case by manually reading all the chat messages. Thus, the precision of our method in this analysis is 100%. Yet, our proposed method missed 1 suspicious community that was identified by the investigators. As a result, the recall of our method in this analysis is $7/8 \approx 88\%$. Our method failed to identify such community due to its infrequent communication. There are two ways to further improve the recall. The first obvious solution is to lower the *min_sup* threshold at the expense of larger number of prominent communities. The second solution is to identify the suspicious information, e.g., e-mail addresses and URLs, from the prominent communities, and then search for such information in the rest of the infrequent communities.

6. Performance analysis

The objective of this section is to study the performance of the prominent community discovery algorithm and the indirect relationship generation algorithm discussed in Section 4. The performance analysis is performed on two real-life datasets. The first dataset is the *Enron* e-mail corpus (Mark and Perrault, 2004). We used *Enron* to analyze the effectiveness of the prominent communities discovery algorithm. Although *Enron* is a *de facto* benchmark dataset used in the field of e-mail forensics, the expected input of our proposed method should be a large collection of files obtained from a file system, not only e-mails. Therefore, to further evaluate the performance, especially the scalability, of our proposed method, we used the hard drive of the first author's personal computer as the second dataset. Throughout the rest of the section, we refer to this dataset as *Filesystem*. We present the analysis results of *Enron* and *Filesystem* in Sections 6.1 and 6.2, respectively.

Table 2
Description of *Filesystem* (40 GB).

File type	Number of files	Size in MB	Percentage
All	43562	40000	100
html	14045	420	1.05
pdf	326	200	0.5
txt	434	1.3	0.00325
xml	995	60	0.15
audio	215	1058	2.645
video	105	840	2.1
MS office	253	66	0.165
other	27022	37354.7	93.38

6.1. The Enron dataset

The *Enron* dataset contains the e-mails of 158 employees in *Enron* Corporation, which was an American energy, commodities, and services company before its bankruptcy. In this experiment, we created two versions of data from the *Enron* dataset, namely *EnronSmall* and *EnronFull*.

EnronSmall contains the e-mails from 30 randomly selected employees, resulting in 48,618 e-mails and 5481 distinct person names. Our proposed method required 16 min to complete the entire process, in which 12 min are spent on extracting all prominent communities for $min\_sup = 8$, 4 min are spent on identifying all indirect relationships, and 3 s are spent on displaying the result. Fig. 5 depicts only a subset of prominent communities identified in *EnronSmall*. Among the 14 prominent communities in the figure, 5 of them are prominent 2-communities and the remaining 9 are prominent 3-communities. When a user clicks on a community, the relevant information, namely other person names, cities, e-mails, phones numbers, and discussed topics, of the community is shown at the bottom of the screen. We inspected the e-mails manually and compared the resulting contact information with the actual content of the messages. The system correctly identified all e-mails and phone numbers without false positives in this case. Fig. 6 depicts a subset of indirect relationships identified in *EnronSmall*. In this particular example, when the mouse was hovered on the link between *john* and {*bush*, *gray*,

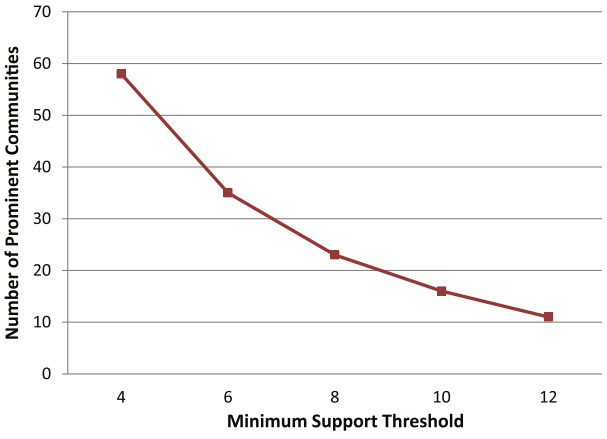

Fig. 8. Number of prominent communities vs. minimum support threshold.

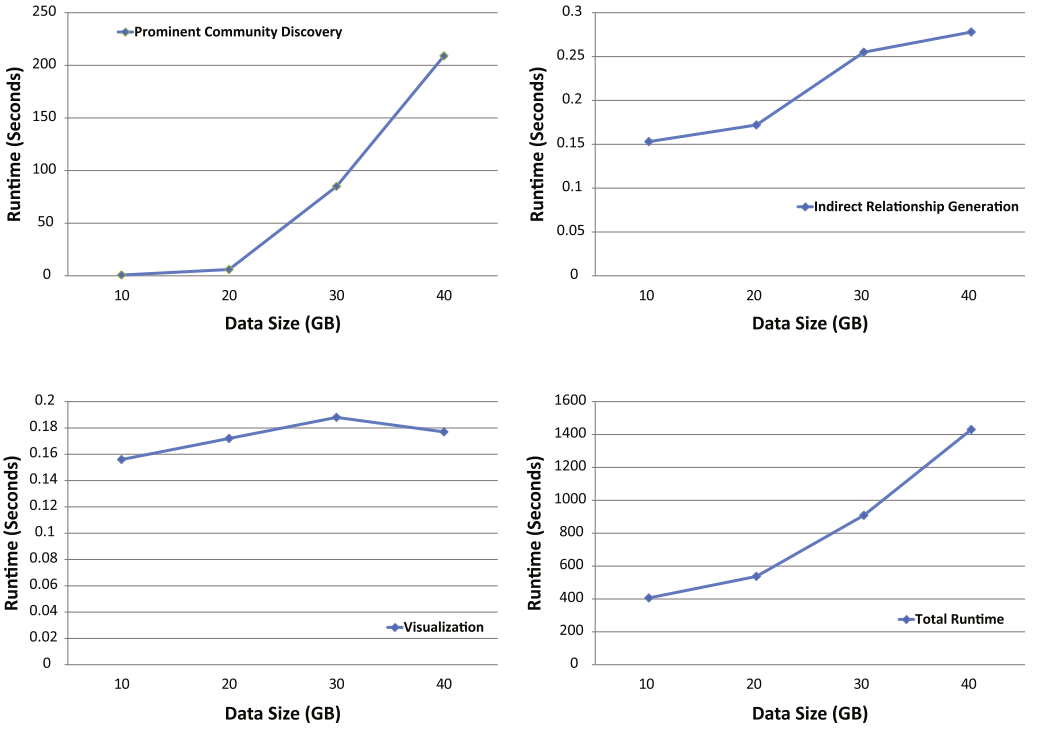

Fig. 9. Scalability: Runtime vs. data size.

davis, *kevin scott*}, the email *jeff.skilling@enron.com popped* up, indicating that *john* was related to the prominent community {*bush*, *gray*, *davis*, *kevin scott*} through the email *jeff.skilling@enron.com*.

EnronFull contains the e-mails from all 158 employees with 2.53 GB of 515,767 e-mails and 108,835 distinct person names. Fig. 7 depicts the number of prominent communities with respect to the number of members (k) in a community for $min\_sup = 250$. The number of prominent communities peaks at $k = 7$. As k increases, a community becomes more difficult to satisfy the minimum support threshold; therefore, the number of prominent communities drops. Our method required 193 min to extract all prominent communities and around 3 s to display the results.

6.2. The Filesystem dataset

The dataset *Filesystem* contains 40 GB of files obtained from the first author's personal computer. Table 2 describes the number of files, size of files, and percentage by file types. Fig. 8 shows the number of prominent communities for $4 \leq min\_sup \leq 12$. As the minimum support threshold increases, the number of prominent communities quickly decreases because the number of documents containing all members in a community decreases very quickly.

Next, we evaluate the scalability of our proposed methods by measuring its runtime. The evaluation is conducted on a PC with Intel 3 GHz Core2 Duo with 3 GB of RAM. Fig. 9 shows the runtime of our proposed methods with respect to the size of the document set which varies from 10 GB to 40 GB with $min\_sup = 8$. The program takes 1430 s to complete the entire process for 40 GB of data, excluding the time spent on reading the document files from the hard drive. As shown in the figure, the total runtime is dominated by prominent community discovery procedure. The runtime of the indirect relationship generation and visualization procedures is negligible with respect to the total runtime.

7. Conclusion

We have proposed an approach to discover and analyze criminal networks in a collection of investigated text documents. Previous studies on criminal network analysis mainly focus on analyzing links between criminals in structured police data. As a result of extensive discussions with a digital forensics team of a law enforcement unit in Canada, we have introduced the notion of prominent criminal communities and an efficient data mining method to bridge the gap of extracting criminal networks information and unstructured textual data. Furthermore, our proposed methods can discover both direct and indirect relationships among the members in a criminal community. The developed software tool has been evaluated by an experienced crime investigator in a Canaidan forensics team and has received positive feedback.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments that greatly helped improve this paper. The research is supported in part by research grants from Le Fonds québécois de la

recherche sur la nature et les technologies (FQRNT) new researchers start-up program, Concordia ENCS seed funding program, and the National Cyber-Forensics and Training Alliance Canada (NCFTA Canada).

References

Agrawal R, Imieliński T, Swami A. Mining association rules between sets of items in large databases. ACM SIGMOD Record 1993;22(2):207–16.

Al-Zaidy R, Fung BCM, Youssef AM. Towards discovering criminal communities from textual data. In: Proc. of the 26th ACM SIGAPP symposium on applied computing (SAC); 2011. TaiChung, Taiwan.

Chen H, Chung W, Xu JJ, Wang G, Qin Y, Chau M. Crime data mining: a general framework and some examples. Computer 2004;37(4):50–6.

Finkel JR, Grenager T, Manning C. Incorporating non-local information into information extraction systems by gibbs sampling. In: Proc. of the 43rd annual meeting on association for computational linguistics (ACL); 2005. p. 363–70.

Friedl JEF. Mastering regular expressions. 3rd ed. O'Reilly Media; 2006.

Geobytes Inc. Geoworldmap, http://www.geobytes.com/; 2003.

Getoor L, Diehl CP. Link mining: a survey. ACM SIGKDD Explorations Newsletter 2005;7(2):3–12.

Hope T, Nishimura T, Takeda H. An integrated method for social network extraction. In: Proc. Of the 15th international conference on world wide web (WWW); 2006. p. 845–6.

Jin W, Srihari RK, Ho HH. A text mining model for hypothesis generation. In: Proc. Of the 19th IEEE international conference on tools with artificial intelligence ICTAI; 2007. p. 156–62.

Jin Y, Matsuo Y, Ishizuka M. Ranking companies on the web using social network mining. In: Ting IH, Wu HJ, editors. Web mining applications in e-commerce and e-services. Studies in computational intelligence, vol. 172. Berlin/Heidelberg: Springer; 2009. p. 137–52.

Mark W, Perrault RC. Enron email dataset, http://www.cs.cmu.edu/~enron/; 2004.

RCFL. Regional computer forensic laboratory annual report 2009. Technical Report. Federal Bureau of Investigation, http://www.rcfl.gov/downloads/documents/RCFL_Nat_Annual09.pdf; 2009.

Rotem N. Open text summarizer, http://libots.sourceforge.net/; 2003.

Skillicorn DB, Vats N. Novel information discovery for intelligence and counterterrorism. Decision Support Systems 2007;43(4):1375–82.

Srinivasan P. Text mining: generating hypotheses from medline. Journal of the American Society for Information Science and Technology 2004;55:396–413.

Xu J, Chen H. Criminal network analysis and visualization. Communications of the ACM 2005;48(6):100–7.

Yang CC, Ng TD. Terrorism and crime related weblog social network: link, content analysis and information visualization. In: IEEE international conference on intelligence and security informatics (ISI); 2007. p. 55–8.

Zhou D, Manavoglu R, Li J, Giles CL, Zha H. Probabilistic models for discovering e-communities. In: Proc. of the 15th international conference on world wide web (WWW); 2006. p. 173–82.