2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology

Mining Criminal Networks from Chat Log

Farkhund Iqbal
College of Information Technology
Zayed University
Abu Dhabi, United Arab Emirates
Email: Farkhund.Iqbal@zu.ac.ae

Benjamin C. M. Fung
CIISE
Concordia University
Montreal, Canada H3G 1M8
Email: fung@ciise.concordia.ca

Mourad Debbabi
CIISE
Concordia University
Montreal, Canada H3G 1M8
Email: debbabi@ciise.concordia.ca

***Abstract*—Cybercriminals exploit opportunities for anonymity and masquerade in web-based communication to conduct illegal activities such as phishing, spamming, cyber predation, cyber threatening, blackmail, and drug trafficking. One way to fight cybercrime is to collect digital evidence from online documents and to prosecute cybercriminals in the court of law. In this paper, we propose a unified framework using data mining and natural language processing techniques to analyze online messages for the purpose of crime investigation. Our framework takes the chat log from a confiscated computer as input, extracts the social networks from the log, summarizes chat conversations into topics, identifies the information relevant to crime investigation, and visualizes the knowledge for an investigator. To ensure that the implemented framework meets the needs of law enforcement officers in real-life investigation, we closely collaborate with the cybercrime unit of a law enforcement agency in Canada. Both the feedback from the law enforcement officers and experimental results suggest that the proposed chat log mining framework is effective for crime investigation.**

***Index Terms*—crime investigation; criminal networks; data mining; frequent patterns; topic mining.**

I. Introduction

Web-based communication, including chat servers, instant messaging systems, and Internet Relay Chat (IRC), provides a convenient medium for broadcasting and exchanging information [1]. Criminals often utilize the anonymous nature of web-based communication to conduct illegal activities. For instance, cyber predators and pedophiles initiate their search for victims in public chat rooms [2].

The content of chat conversation may directly or indirectly reveal the social networks, activities, preferences, and lifestyles of the participants. In real-life investigations, sometimes a crime investigator has access to archived chat logs in confiscated computers and in public chat servers. Yet, analyzing and identifying evidence relevant to a criminal case from a large volume of chat conversations could be challenging and time-consuming. Currently, most investigators rely on the simple search functionality in off-the-shelf forensic software tools or in desktop search engines, such as Google Desktop, to extract relevant information. Yet, this approach suffers from three major shortcomings: (1) Existing search tools fail to reveal the social networks and activities of the suspects. (2) Most searching and network mining tools are designed for structured documents and, therefore, fail to analyze unstructured document such as chat log or instant messages. (3) Some other tools focus on analyzing header and network level information, such as IPs, protocols, and path traveled by network packets and can not be used for analyzing the content or body of online messages.

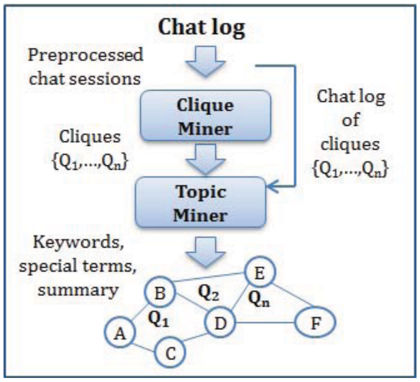

Chat log
Preprocessed chat sessions
Clique Miner
Cliques {Q₁,...,Qₙ}
Chat log of cliques {Q₁,...,Qₙ}
Topic Miner
Keywords, special terms, summary
A
B
C
D
E
F
Q₁
Q₂
Qₙ

Fig. 1. Framework overview

To address these shortcomings in existing forensic search tools, we propose a data mining framework to extract social networks from a given chat log, summarize the discussed topics in every identified group, and allow the investigator to perform a search on the results. The objective is to collect from a chat log intuitive and interpretable evidence that facilitates the investigative process, especially in the early stage when an investigator has few clues to start with. Figure 1 presents an overview of the proposed framework that consists of three modules: *clique miner*, *topic miner*, and *information visualizer*. The clique miner first extracts the entities from a given chat log. Depending on the context of the investigation, the term *entity* is a general concept that can refer to the name of a person, an organization, a phone number, or a physical address. To ease the discussion, we assume the term refers to the name of a person. Next, the *clique miner* identifies the social networks, called *cliques*, based on the co-occurrence frequencies of the entities in chat sessions. Next, the *topic miner* extracts the topics discussed in the chat sessions of each clique. Finally, the *information visualizer* displays the cliques at different levels of abstraction as an interactive graph in which the nodes represent the entities and the links represent the cliques, labeled with the conversation summary, keywords, and domain-specific terms of the selected clique.

The contributions of this study are summarized as follows:

978-0-7695-4880-7/12 $26.00 © 2012 IEEE
DOI 10.1109/WI-IAT.2012.68

- *Social networks mining from unstructured data:* Most of the existing works in crime data mining focus on discovering knowledge in structured data [3]. In contrast, our study focuses on extracting useful information from a given chat log consisting of a collection of unstructured chat sessions written in free-text. Unlike police narrative reports, which usually follow a specific template or structure, chat logs are more challenging to analyze due to their limited size, informal composition style, and the presence of spelling and grammatical mistakes.
- *Customized notion of clique for criminal networks mining:* The traditional notion of a social group usually considers the number of direct interactions among the participants, for example, the number of chat sessions, e-mails, or blog postings exchanged. After an extensive discussion with law enforcement officers, we find that simply counting the number of direct interactions is too limited and often results in missing some key entities in a clique. Thus, we define a customized notion of clique for criminal network mining. In the context of chat log mining, a clique is a set of entities that appear together in some minimum number of chat sessions, even if they do not chat directly.
- *Topic identification without a priori knowledge:* Most of the existing topic classification and identification techniques assume an investigator has access to a list of predefined topics with sample documents in order to train a classification model. Yet, these kinds of data are seldom available in real-life crime investigations. Our approach does not require such a priori information and can extract important keywords or common topics based solely on the content of the chats in question. Thus, the topics identified by our method can better represent the main ideas of the underlying chat sessions.

The rest of the paper is organized as follows. Section II discusses the related tools and literature. Section III formally defines the problem of clique mining and topic analysis in the context of criminal networks mining. Section IV presents our proposed chat log mining framework. Section V presents the experimental results on a carefully synthesized dataset. Section VI concludes the paper with a summary and directions for future work.

II. Related Works and Tools

Understanding the dynamics behind the relationships between criminals can help an investigator identify suspects and understand criminal activities [4]. More specifically, analyzing social networks can help identify subnetworks, actors, roles of each actor, and patterns of communication among the actors [5]. For instance, Stolfo and Hershkop [6] demonstrated the feasibility of extracting social networks from an e-mail corpora and revealing hidden information such as the communication paths among the actors, the number of communities in the corpus, and the key actors and topics bridging the communities. A social network is often represented as a graph in which nodes represent individuals or actors and links represent relationships among the actors. In the literature, different approaches have been developed for extracting a social network from both structured and unstructured textual data.

Culotta et al. [7] proposed an approach to identify personal names in e-mail content and then to look for additional information, including contact information and topics of interests, from the web using *conditional random fields*, a probabilistic model that performs well on similar language processing tasks [8]. Furthermore, their method uses the contact information and the topic (represented by keywords) to identify related people sharing the same information, thus building a social network.

Chen et al. [3] developed a crime data mining framework by integrating state-of-the-art data mining techniques such as link analysis, association rule mining, and knowledge discovery. The framework has the capability of identifying different kinds of crimes. Chau et al. [9] applied different *link analysis* techniques on the Tucson, Arizona, police department database to identify covert association between crime entities. The proposed techniques, including shortest path algorithm, co-occurrence analysis, and a heuristic approach, are successful in identifying associations between entities.

The techniques proposed in some of the studies focus on analyzing structured data such as police reports and newswire articles. Social networks are often built from header information such as an IP address and e-mail address rather than from the body content of online documents. The relation between the nodes (e.g., web pages, web logs, or e-mail addresses) is based solely on their communication frequency, measured in terms of in-link, out-link, and centrality [10]. However, in real life two entities may be related even if they never communicated with each other directly. To extract such relationships it is imperative to analyze the content of online conversation. Once the social networks of a person are extracted from his/her textual conversation, they may reveal the different aspects of his/her social life, e.g., job, family, friends, or criminal activities. For this purpose, it is important to identify the topic discussed in each social network.

Recently, Al-Zaidy et al. [11] proposed a social network mining method to extract social groups from textual files. However, their method does not analyze the interactions among the authors nor the discussed topic as we do in this paper. Considering these two factors is important for crime investigation on chat logs.

III. Problem Statement

Suppose an investigator has seized a computer from a suspect S. Let Φ be the chat log obtained from the computer from some commonly used instant-messaging system such as Windows Live Messenger, Yahoo! Messenger, or IRC. Typically, a chat log consists of a set of chat sessions, where each session contains a set of text messages exchanged between suspect S and the chat users who appear in the friend list of S. The *problem of criminal clique mining* is to discover the communities (i.e., cliques) the suspect S in

Φ is actively involved in, identify the relationships among the members in the cliques, and extract the topics that bring members of a clique together. The problem can be divided into two subproblems: *clique mining* and *topic analysis*.

A. Subproblem: Clique Mining

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

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

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

A chat log Φ is a collection of chat sessions $\{\phi_1, \ldots, \phi_p\}$. Let $E(\Phi) = \{\epsilon_1, \ldots, \epsilon_u\}$ denote the universe of all entities identified in Φ. Let $E(\phi_i)$ denote the set of entities identified in a chat session ϕ_i, where $E(\phi_i) \subseteq E(\Phi)$. For example, $E(\phi_5) = \{\epsilon_4, \epsilon_5, \epsilon_7\}$ in Table I. Let $Y \subseteq E(\Phi)$ be a set of entities called *entityset*. A session ϕ_i contains an entityset Y if $Y \subseteq E(\phi_i)$. An entityset that contains k entities is called a k*-entityset*. For example, the entityset $Y = \{\epsilon_3, \epsilon_6, \epsilon_7\}$ is a 3-entityset. The *support* of an entityset Y is the percentage of chat sessions in Φ that contain Y. An entityset Y is a clique in Φ if the support of Y is greater than or equal to some user-specified minimum support threshold.

Definition 3.1 (Clique): Let Φ be a collection of chat sessions. Let $support(Y)$ be the percentage of sessions in Φ that contain an entityset Y, where $Y \subseteq E(\Phi)$. An entityset Y is a clique in Φ if $support(Y) \geq min\_sup$, where the minimum support threshold min_sup is a real number in an interval of [0, 1]. A clique containing k entities is called a k*-clique*. ■

TABLE I
VECTORS OF ENTITIES REPRESENTING CHAT SESSIONS

Chat session	Identified entities
ϕ_1	$\{\epsilon_2, \epsilon_5, \epsilon_7, \epsilon_9\}$
ϕ_2	$\{\epsilon_2, \epsilon_5, \epsilon_7\}$
ϕ_3	$\{\epsilon_2, \epsilon_5\}$
ϕ_4	$\{\epsilon_1, \epsilon_5, \epsilon_7\}$
ϕ_5	$\{\epsilon_4, \epsilon_5, \epsilon_7\}$
ϕ_6	$\{\epsilon_3, \epsilon_6, \epsilon_8\}$
ϕ_7	$\{\epsilon_4, \epsilon_5, \epsilon_8\}$
ϕ_8	$\{\epsilon_3, \epsilon_6, \epsilon_8\}$
ϕ_9	$\{\epsilon_2, \epsilon_5, \epsilon_8\}$
ϕ_{10}	$\{\epsilon_1, \epsilon_5, \epsilon_7, \epsilon_8, \epsilon_9\}$

Example 3.1: Consider Table I. Suppose the user-specified threshold $min\_sup = 0.3$, which means that an entityset Y is a clique if at least 3 out of the 10 sessions contain all entities in Y. Similarly, $\{\epsilon_4, \epsilon_5\}$ is not a clique because it has support 2/10 = 0.2. $\{\epsilon_2, \epsilon_5\}$ is a 2-clique because it has support 4/10 = 0.4 and contains 2 entities. Likewise, $\{\epsilon_5, \epsilon_8\}$ is a 2-clique with support 3/10 = 0.3. ■

Definition 3.2 (Clique mining): Let Φ be a collection of chat sessions. Let min_sup be a user-specified minimum support threshold. The subproblem of *clique mining* is to efficiently identify *all* cliques in Φ with respect to min_sup. ■

B. Subproblem: Topic Analysis

According to Canadian law enforcement officers, some criminal cases they have encountered involve thousands of chat users in the Windows Live Messenger chat log on a single machine. Consequently, hundreds of cliques could be discovered in the chat log. The revealed cliques reflect different social aspects of the suspect, including family, friends, work, and religion. To identify cliques related to criminal activities the investigator has to analyze the content of the chat sessions of each clique. The subproblem of topic analysis is to extract the topics that reflect the meaning of the underlying chat conversations.

Definition 3.3 (Topic analysis): Let Q be a set of cliques discovered in Φ according to Definition 3.2. Let $\Phi(Q_i) \subseteq \Phi$ be the set of chat sessions contributing to the support of a clique $Q_i \in Q$. Note that the same chat session may contribute to multiple cliques. The subproblem of *topic analysis* is to extract a set of common topics, denoted by $KW(X)$ and S(X), for each discovered clique $Q_i \in Q$. The topics bring the group of entities to form a clique. ■

IV. Proposed Approach

Figure 2 depicts an overview of our proposed framework that consists of three components: *clique miner*, *topic miner*, and *information visualizer*. *Clique miner* identifies all the cliques and their support from the given chat log. *Topic miner* analyzes the chat sessions of each identified clique and extracts the common topics of the conversations. *Information visualizer* provides a graphical interface to allow the user to interactively browse cliques at different abstraction levels. A detailed screen

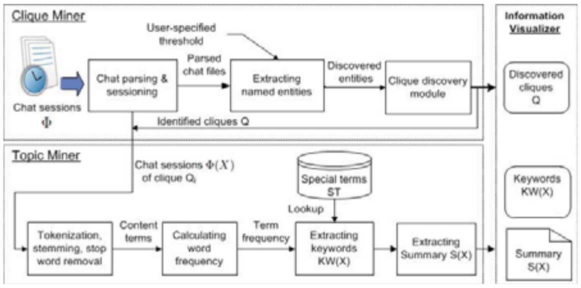


Fig. 2. Components of the proposed criminal information mining framework

shot is provided in Section V. Refer to our technical report [12] for more details. The rest of this section focuses on the clique miner and topic miner.

A. Clique Miner

The process of clique mining consists of three steps:

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

(2) *Extracting entities:* Next, we employ an existing statistical-based Named Entity Recognition (NER) tools[1] to extract entity names from each chat session. In this study, we assume an entity is a person, but in real-life application, an entity can also be an organization, location, phone number, or website [13]. Other NER tools can be employed if the document files contain non-English names, as NER is not the focus of this study. The next step, clique mining, operates on a data table consisting of records of entities. Each record contains the entities identified in a chat session.

(3) *Mining cliques:* An entityset Y is any combination of entities identified in the chat log, and it is a clique if its support is equal to or greater than a given threshold. A naive approach is to enumerate all possible entitysets and identify the cliques by counting the support of each entityset in Φ. Yet, in case the number of identified entities $|E(\Phi)|$ is large, it is infeasible to enumerate all possible entitysets because there are $2^{|E(\Phi)|}$ possible combinations. We modify the Apriori algorithm [14], originally designed to extract frequent patterns from transaction data, to efficiently extract all cliques from Φ.

Algorithm 1 Clique Miner

```
Input: Chat log $\Phi$
Input: Minimum support threshold $min\_sup$
Output: Cliques $Q = \{Q_1 \cup \ldots \cup Q_k\}$
Output: Chat sessions $\Phi(X)$, $\forall X \in Q$
$Q_1 \leftarrow \{\epsilon \mid \epsilon \in E(\Phi) \Lambda\ support(\{\epsilon\}) \geq min\_sup\}$;
for ($k = 2$; $Q_{k-1} \neq \emptyset$; $k$++) do
    $Candidates_k \leftarrow Q_{k-1} \bowtie Q_{k-1}$;
    for all entityset $Y \in Candidates_k$ do
        if $\exists Y' \subset Y$ such that $Y' \notin Q_{k-1}$ then
            $Candidates_k \leftarrow Candidates_k - Y$;
        end if
    end for
    $\Phi(X) \leftarrow \emptyset$, $\forall X \in Candidates_k$;
    for all chat session $\phi \in \Phi$ do
        for all entityset $X \in Candidates_k$ do
            if $X \subseteq E(\phi)$ then
                $\Phi(X) \leftarrow \Phi(X) \cup \phi$;
            end if
        end for
    end for
    $Q_k \leftarrow \{X \mid X \in Candidates_k \Lambda |\Phi(X)| \geq min\_sup\}$;
end for
$Q = \{Q_1 \cup \ldots \cup Q_k\}$;
return $Q$ and $\Phi(X)$, $\forall X \in Q$;
```

Recall that $E(\Phi)$ denotes the universe of all entities in Φ, and $E(\phi_i)$ denotes the set of entities in a session $\phi_i \in \Phi$, where $E(\phi_i) \subseteq E(\Phi)$. Our proposed Clique Miner is a level-wise iterative search algorithm that uses the k-cliques to explore the $(k+1)$-cliques. The generation of $(k+1)$-cliques from k-cliques is based on the downward closure property [14].

Property 4.1 (Downward closure property): All nonempty subsets of a clique are also cliques because $support(Y') \geq support(Y)$ if $Y' \subseteq Y$. ■

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

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

The following example shows how to efficiently extract all

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

frequent patterns.

Example 4.1: Consider Table I with $min\_sup = 0.3$. First, identify all the entities by scanning the table once to obtain the support of every entity. The entities having support ≥ 0.3 are 1-cliques $Q_1 = \{\{\epsilon_2\}, \{\epsilon_5\}, \{\epsilon_7\}, \{\epsilon_8\}\}$. Then join Q_1 with itself, i.e., $Q_1 \bowtie Q_1$, to generate the candidate set $Candidates_2 = \{\{\epsilon_2, \epsilon_5\}, \{\epsilon_2, \epsilon_7\}, \{\epsilon_2, \epsilon_8\}, \{\epsilon_5, \epsilon_7\}, \{\epsilon_5, \epsilon_8\}, \{\epsilon_7, \epsilon_8\}\}$ and scan the table once to obtain the support of every entityset in $Candidates_2$. Next, identify the 2-cliques $Q_2 = \{\{\epsilon_2, \epsilon_5\}, \{\epsilon_5, \epsilon_7\}, \{\epsilon_5, \epsilon_8\}\}$. Similarly, perform $Q_2 \bowtie Q_2$ to generate $Candidates_3 = \{\epsilon_5, \epsilon_7, \epsilon_8\}$ and determine $Q_3 = \emptyset$. Finally, the algorithm returns Q_2 and the associated chat sessions of every clique in Q_2. ■

B. Topic Miner

This phase is to analyze the chat sessions and summarize the content into some high-level topics to facilitate effective browsing in the visualization phase. The topic miner extracts common topics from the set of associated chat sessions $\Phi(X)$ of every clique $X \in Q$ identified by the clique mining algorithm. It is important to extract the topics that are commonly discussed in a subnetwork, representing a specific social group. This step is imperative for identifying the type of a social group, indicating whether it is legitimate or malicious. Identifying the topic of online messages is difficult because they are unstructured and are usually written in para language. The abbreviations, special symbols, and visual metaphors used in malicious messages convey special meanings and are meaningful in some specific context. Specifically, the topic miner extracts two notions from $\Phi(X)$: *Keywords* are either frequent words or domain-specific words extracted from $\Phi(X)$. *Summary* containing key sentences, is extracted from the chat sessions $\Phi(X)$ of clique $X \in Q$.

We first applied some standard text mining preprocessing procedures, including tokenization, stop-word removal, and stemming [15]. After preprocessing, each chat session is represented as a vector of terms. For every clique $X \in Q$, we extract the keywords from $\Phi(X)$, prepare a summary of the chat conversation based on the extracted keywords and domain-specific terms. We elaborate these two steps as follows:

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

- Some special terms that may not frequently appear are important for crime investigation. For instance, certain crime-relevant street terms such as marijuana, heroin, or opium are relevant and therefore require more attention even though they may appear only once. To identify such special terms, we allow the investigator to specify a list of special terms, denoted by ST. In our implementation the terms are collected from different law enforcement agencies and online sources.[2]
- A term is important in $\Phi(X)$ if it frequently appears in the chat session $\Phi(X)$ of clique $X \in Q$ but not frequently in chat session $\Phi(Y)$ of other clique $Y \in Q$, where $X \neq Y$. Intuitively, these terms can help differentiate the topic of one clique from others. To identify them, we compute the $tf - idf$ of every term and add the top α of them to $KW(X)$, where α is a user-specified threshold.

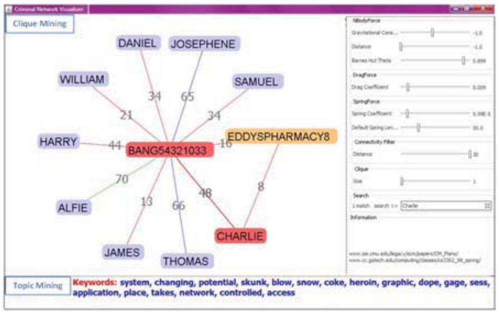
(2) *Extracting summary:* Summary of the chat conversation $\Phi(X)$ of clique $X \in Q$ is extracted by employing a popular text summarization technique, presented in [16]. The sentences containing the keywords are key sentences, which constitute the summary of chat conversation.

V. Experiments and Discussion

The experimental evaluation serves three objectives: (1) to evaluate if the cliques extracted by the clique miner represent a meaningful group of individuals in the real world and to measure the effect of a minimum support on the number of cliques; (2) to evaluate whether or not the topic miner can precisely identify the discussed topic from the chat conversation of each extracted clique; and (3) to investigate whether or not the proposed topic mining technique can help in identifying the different social groups of a person.

To evaluate the performance of our proposed method, we have created a synthetic dataset. The synthetic dataset contains some *MSN* chat logs created by our research team based on information extracted from an anonymous source.

A. *MSN Chat Log*



Keywords: system, changing, potential, skunk, blow, snow, coke, heroin, graphic, dope, gage, sess, application, place, takes, network, controlled, access

Fig. 3. A sample screen shot of the developed framework

Figure 3, a screen shot of our developed framework, shows the cliques identified from the *MSN* chat log. This figure shows ten cliques, each containing two or three entities. The central node, which is part of all other cliques, represents the *MSN* user (i.e., the suspect) of the confiscated computer and the remaining nodes represent the entities associated with the suspect. The arcs connecting the entities indicate the relationships between the entities. Through the user interface, a user can highlight a clique to display more information

[2] http://www.whitehousedrugpolicy.gov/streetterms/

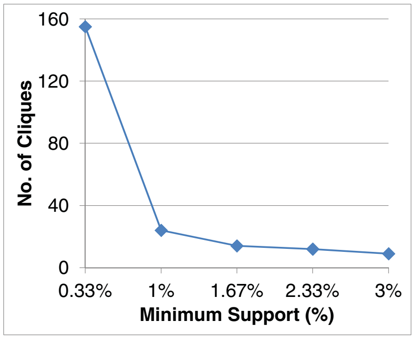


Fig. 4. Impact of minimum support on number of cliques

about its members. For example, the clique containing entities *BANG54321033*, *EDDYSPHARMACY8*, and *CHARLIE* is interesting as the chat conversation of its member contains drug-related terms, e.g., grass, skunk, and snow. We manually compared the extracted entities and the discovered cliques with the textual content of the given chat sessions. We found that more than 80% of the cliques are correctly identified, with a few false positive cases. Figure 4 depicts the number of cliques with respect to the minimum support threshold. As the threshold increases from 0.33% to 3.33%, the number of cliques quickly decreases from 155 to 8.

We also evaluate the topic mining functionality of the presented framework. The concept miner retrieves the chat log of each clique, discovered in the clique mining step, and extract the keywords, domain-specific terms, and conversation summary from each chat collection separately. Figure 3 visualizes the extracted cliques and the topic mining results associated with each clique. The drill-down and roll-up capabilities of the framework allow the user to browse the cliques and the summary of their conversation.

We found the topic analysis summary of the chat log belonging to the clique of *BANG54321033*, *EDDYSPHARMACY8*, and *CHARLIE* interesting. The extracted keywords, such as blow, snow, coke, dope, and gage, are street terms used to represent cocaine, a narcotic. The topic miner also identifies other words such as *system*, *changing*, and *potential* as keywords, due to the high frequency of occurrences. By comparing the extracted keywords and the related keywords with the content of associated chat sessions, we conclude that the topic miner can correctly identify the topic of online messages.

VI. Conclusion

In this paper, we have developed a criminal information mining framework for extracting forensically relevant information from suspicious online messages. The framework takes online messages as input and identifies a set of cliques, together with the discussed topics in the chat conversation of each clique, as output. The experimental results on a carefully synthesized dataset suggest that the proposed framework can precisely identify the pertinent cliques and the perceived meaning of the messages exchanged between members of the cliques. The work is developed under a close collaboration with a cyber forensics team in Canada. The effectiveness of the method has been confirmed by experienced crime investigators.

Acknowledgments

This work was done while the first author was a research fellow in the Concordia Institute for Information Systems Engineering, Concordia University. The research is supported in part by the National Cyber-Forensics and Training Alliance Canada (NCFTA Cda) and the new researchers start-up program from Le Fonds québécois de la recherche sur la nature et les technologies (FQRNT). The authors would like to thank Mr. Mirza Ahmed for his contributions in the early stage of the project.

References

[1] T. Kucukyilmaz, B. B. Cambazoglu, F. Can, and C. Aykanat, "Chat mining: predicting user and message attributes in computer-mediated communication," *Information Processing and Management*, vol. 44, no. 4, pp. 1448–1466, 2008.

[2] N. Pendar, "Toward spotting the pedophile telling victim from predator in text chats," *IEEE Internet Computing*, pp. 235–241, 2007.

[3] H. Chen, W. Chung, J. J. Xu, G. Wang, Y. Qin, and M. Chau, "Crime data mining: A general framework and some examples," *Computer*, vol. 37, no. 4, 2004.

[4] J. S. McIllwain, "Organized crime: A social network approach," *Crime, Law and Social Change*, vol. 32, pp. 301–323, 1999.

[5] C. Yang, N. Liu, and M. Sageman, "Analyzing the terrorist social networks with visualization tools," in *Proc. of Intelligence and Security Informatics*, San Diego, CA, 2006, pp. 331–342.

[6] S. J. Stolfo and S. Hershkop, "Email mining toolkit supporting law enforcement forensic analyses," in *Proc. of the National Conference on Digital Government Research*, 2005, pp. 221–222.

[7] A. Culotta, R. Bekkerman, and A. Mccallum, "Extracting social networks and contact information from Email and the Web," in *Proc. of the 1st Conference on Email and Anti-Spam*, Mountain View, California, USA, 2004, pp. 1–8.

[8] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. of 18th International Conference on Machine Learning*, San Francisco, CA, 2001, pp. 282–289.

[9] M. Chau, J. Schroeder, J. Xu, and H. Chen, "Automated criminal link analysis based on domain knowledge," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 6, pp. 842–855, 2007.

[10] M. E. J. Newman, S. Forrest, and J. Balthrop, "Email networks and the spread of computer viruses," *Physical Review E*, vol. 66, no. 3, 2002.

[11] R. Al-Zaidy, B. C. M. Fung, A. Youssef, and F. Fortin, "Mining criminal networks from unstructured text documents," *Digital Investigation*, vol. 8, no. 3-4, pp. 147–160, February 2012.

[12] F. Iqbal, B. C. M. Fung, and M. Debbabi, "Mining criminal networks from chat log," Concordia University, Tech. Rep. 20120509, May 2012, http://www.ciise.concordia.ca/∼fung/pub/TR20120509.pdf.

[13] E. Alfonseca and S. Manandhar, "An unsupervised method for general named entity recognition and automated concept discovery." in *Proc. of the International Conference on General WordNet*, Mysore, India, 2002.

[14] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proc. of ACM SIGMOD international conference on Management of data*. Washington, DC: ACM, 1993, pp. 207–216.

[15] C. D. Paice, "Another stemmer," *SIGIR Forum*, vol. 24, no. 3, pp. 56–61, 1990.

[16] V. A. Yatsko and T. N. Vishnyakov, "A method for evaluating modern systems of automatic text summarization," *Automatic Documentation and Mathematical Linguistics*, vol. 41, no. 3, pp. 93–103, 2007.