IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 9, 2018, accepted December 28, 2018, date of publication January 9, 2019, date of current version March 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2891694

Wordnet-Based Criminal Networks Mining for Cybercrime Investigation

FARKHUND IQBAL[1], BENJAMIN C. M. FUNG[2], (Senior Member, IEEE), MOURAD DEBBABI[3], RABIA BATOOL[1], AND ANDREW MARRINGTON[1]

[1]College of Technological Innovation, Zayed University, Abu Dhabi 144534, United Arab Emirates
[2]School of Information Studies, McGill University, Montreal, QC H3A 1X1, Canada
[3]Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC H3G 1M8, Canada

Corresponding author: Farkhund Iqbal (farkhund.iqbal@zu.ac.ae)

This work was supported in part by the Research Incentive Fund under Grant R15048, and in part by the Research Cluster, Zayed University, United Arab Emirates, under Grant R16083.

ABSTRACT Cybercriminals exploit the opportunities provided by the information revolution and social media to communicate and conduct underground illicit activities, such as online fraudulence, cyber predation, cyberbullying, hacking, blackmailing, and drug smuggling. To combat the increasing number of criminal activities, structure and content analysis of criminal communities can provide insight and facilitate cybercrime forensics. In this paper, we propose a framework to analyze chat logs for crime investigation using data mining and natural language processing techniques. The proposed framework extracts the social network from chat logs and summarizes conversation into topics. The crime investigator can use information visualizer to see the crime-related results. To test the validity of our proposed framework, we worked in a joint effort with the cybercrime unit of a Canadian law enforcement agency. The experimental outcomes on real-life data and feedback from the law enforcement officers suggest that the proposed chat log mining framework meets the need for law enforcement agencies and is very effective for crime investigation.

INDEX TERMS Data mining, crime investigation, criminal communities, clustering algorithms, WordNet.

I. INTRODUCTION

Communication trends of the digital era, including chat servers and Instant Messaging (IM) systems, offer a convenient method for sharing information [16]. Criminals exploit these opportunities to perform illicit activities. The National Center for Missing and Exploited Children [32] reported that one out of every seven children in the United States faces unwanted online sexual solicitations. Drug dealers use chat rooms for drug trafficking. Terrorists and hate groups use social media to promote their ideology [10], [50].

Most of these chat and IM systems have an archive feature that stores all conversations for later reference. If the investigator gets access to the archived conversations in confiscated computers or in public chat servers, those conversations can be helpful in crime investigation. The content of online communication can reveal the lifestyles of the participants, their social networks, activities, and preferences. However, manually analyzing a large volume of chat conversations to find evidence related to a criminal case is very tedious and time-consuming. Most investigators use traditional search methods in forensic software tools or in desktop search engines to crawl and extract relevant data. This method has three noteworthy limitations: (1) Existing search tools may find documents and sentences related to the search terms, but they do not provide information related to a suspect's social networks and activities. (2) The simple string-matching method is not suitable for investigation of crime cases because drug dealers seldom use the terms *"drug"* and *"cocaine"* in their conversation. (3) The experience and prior knowledge of the person performing the search are crucial for good quality search results; otherwise, a search can often result in irrelevant, incomplete, or inconsistent information.

To address the limitations of existing forensic search tools, we propose a framework that extracts communities from a given chat log and uses agglomerative clustering to find and summarize the topics of interest in the identified communities. Crime investigators can perform a search query and see the results in the designed visualizer. The purpose of this data mining framework is to collect instinctive and interpretable

The associate editor coordinating the review of this manuscript and approving it for publication was Manik Sharma.

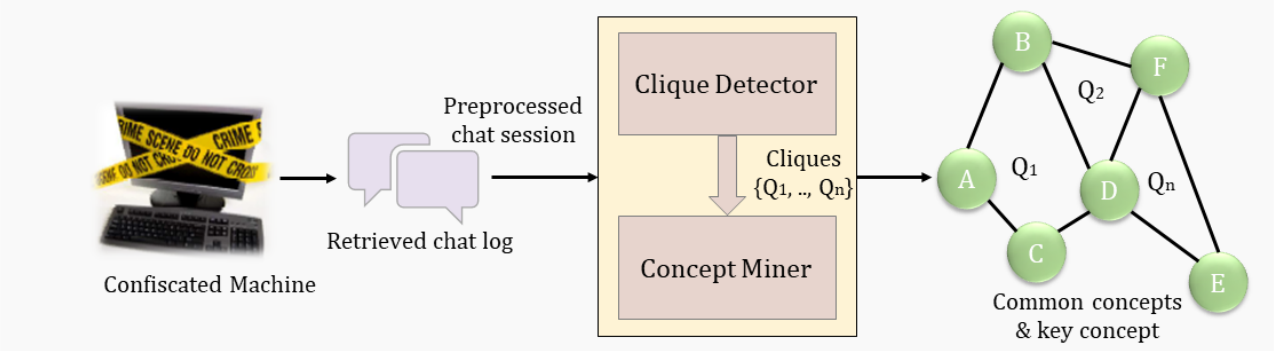

FIGURE 1. Overview of the proposed framework.

evidence from a chat log to simplify and facilitate the investigation process, especially in the initial stage when there are not enough indications for the investigator to start with. An overview of the proposed framework is presented in Figure 1. There are three main modules of the framework: *Clique Detector*, *Concept Miner*, and *Information Visualizer*.

The *Clique Detector* identifies the cliques (communities) in the chat log. For this purpose, it first recognizes all the mentioned entities in the given chat log. In the context of investigation, the term *entity* can be the name of a person, an organization, a phone number, or a physical address. For simplicity, we assume that the entity refers to the person's name. After extracting entities, the *Clique Detector* uses the co-occurrence frequencies of the entities in chat sessions and identifies the communities, called *cliques*.

Next, the *Concept Miner* processes a chat log of each clique and extracts the concepts that represent the discussed topics. For this purpose, it identifies important terms based on their frequency in the text. Each identified important term is then mapped to a corresponding concept in the WordNet [19] that is used to create a hierarchy of concepts and to represent relationships between them. A customized version of agglomerative hierarchical clustering is applied to form groups of concepts holding strong cohesive relationships among terms. The top node of the concept hierarchy is the main topic of the conversation.

Finally, the *Information Visualizer* displays the identified cliques as an interactive graph in which the nodes represent the recognized entities, and the edges represent the identified cliques. The Information Visualizer also shows the summary, keywords, and concepts of the selected clique from the graph.

The major contributions of this study are:

1) *Social communities mining from unstructured textual data:* Most of the research in the domain of crime investigation uses structured data for knowledge discovery [12]. In this study, our aim is to mine chat logs to extract important and valuable information. A chat log consists of several chat sessions written in natural language. Chat conversations are sometimes very short and are often prone to typos and grammatical mistakes; therefore, accurate information extraction is challenging.
2) *Customized notion of a clique for criminal communities mining:* The traditional methods of network analysis generally count the number of direct interactions between the members, e.g., the count of e-mails exchanged. After a discussion with law enforcement officers, we discovered that simply considering the number of direct interactions is insufficient and can lead to outcomes with missing or incomplete information. Thus, a customized notion of a clique for criminal communities mining is defined in this study. In the context of chat log mining, the entities that frequently appear together in the chat sessions are considered as a clique even though they do not have a direct conversation.
3) *Concept identification without any prior knowledge:* Many existing topic identification techniques require investigators to have training data to train a classification model. However, in the case of a crime investigation, this kind of data is not easily available. Our proposed framework does not need any prior information or training data and can identify key concepts (or topics) based on the content of the chats, with the help of a lexical database WordNet [19]. Furthermore, the concept extraction process relies on the semantic similarity of words rather than their frequencies. Hence, our technique is able to identify the most relevant concepts that can better represent and summarize the information contained in the chat sessions.
4) *Flexiblity to adopt domain knowledge:* Our proposed criminal communities mining framework allows an investigator to incorporate domain knowledge with the goal of improving the analysis process. Domain knowledge can be a word taxonomy used in malicious online conversations that represents street terms of certain crimes.

The remainder of this paper is organized as follows. Section II provides some background information about WordNet and Named-Entity Recognition. Section III examines the literature that closely aligns with our study. Section IV highlights the problem of cliques and concept mining for the criminal network. The proposed framework is presented in Section V. Section VI discusses the evaluation

and experimental results on some real-life datasets. Conclusion and direction for future work are provided in Section VII.

II. BACKGROUND INFORMATION

This section defines a *named entity* and explains its importance for communities mining. Further, we discuss the structure of WordNet and its importance for the proposed approach.

A. NAMED-ENTITY RECOGNITION

The term *Named-Entity Recognition*, or NER, is used in information extraction to identify information associated with an entity [13]. An entity is a pre-specified element that can be a person, place, or organization. The attribute of an entity can be a phone number, e-mail, URL, birth date, vehicle number, or ID number [20]. NER has been successfully utilized in criminal information mining. Chen *et al.* [11] utilized a neural network-based NER to extract the identities of criminals from police reports and suspicious documents. Similarly, Shabat and Omar [52] presented a NER system to extract a crime named entity and crime type from crime documents. Cybercriminals sometimes hide their true identities by using identity deception strategies. To detect masqueraded criminal identities, Wang *et al.* [31] presented a record-linkage automated deception detection algorithm.

B. WORDNET

WordNet is a large lexical database of the English language that was developed at Princeton University [19], and it is freely available.[1] It contains more than 120,000 concepts, including nine *noun* hierarchies (80,000 concepts) and 554 *verb* hierarchies (13,500 concepts) [23]. WordNet is specifically designed for text mining and artificial intelligence applications. Unlike a traditional dictionary that contains the spelling, pronunciation, synonyms, and antonyms of words, WordNet identifies the semantic relationship between word senses. A synset (synonym set), representing a distinct concept, forms the basic building block of WordNet. Different language artifacts, i.e., noun, verb, adjective, and adverb, are organized into networks of synsets that are semantically and lexically related to each other.

In the WordNet semantic network, nouns and verbs are related to each other in *is-a* hierarchy without crossing the part-of-speech boundary. For instance, two nouns, *fruit* and *mango*, or two verbs, *talk* and *speak*, are related to each other vertically. The subordination relation between lexicalized concepts in the *is-a* hierarchy is called *hyponymy*. Hypernyms are abstract concepts that are stored at the higher level, and the more specific/specialized concepts, known as hyponyms, are stored at the lower level in the hierarchy. For example, in the noun pair, *"drug"* and *"cocaine"*, the word *drug* is the hypernym while the word *cocaine* is the hyponym.

In addition to the *is-a* relationship between verbs and nouns, WordNet identifies the relationship between other language artifacts, such as adverbs and adjectives. For example, *meronymy* is used to represent a whole-part relationship, e.g., a motorcar 'has an' engine or an engine 'is a part of' a motorcar. In this example, the motorcar is the holonym while the engine is the meronym.

[1] http://wordnet.princeton.edu/

III. RELATED WORK

Dynamics behind a criminal's relationship can be very helpful in identifying suspects and understanding the activities and interests of criminals [18]. For any criminal event, crime investigators must consider and retrieve criminal network information [49]. The investigation related to social networks can reveal hidden subnetworks, actors, their roles, and communication pattern [35]. The investigation team can use the information of evidential value from a chat log to recreate events. It can be helpful in identifying information such as the physical location of suspects, identity, transactional information, and information about the victim [48].

Chen *et al.* [12] presented a framework that uses data mining techniques to extract criminal relations from the police department's incident summaries. They used the concept-space approach to determine the relationships between pairs of criminals. They also identified subgroups and key members in the criminal networks. Yang and Ng [36] used a topic-specific exploration mechanism to extract semantics, topics, and criminal networks from blogging websites.

A graph is usually used to represent the social network in which the nodes represent the individuals, and the relationships between the individuals are indicated by edges. Researchers have proposed and developed many different approaches to extract and analyze social networks from both structured and unstructured textual data. Stolfo and Hershkop [29] developed an email mining toolkit for detectives and analysts to visualize hidden information such as the number of communities, the communication paths, and users' behaviors. This toolkit highlights the suspicious emails account and helps analysts find proxy accounts, security breaches, and spam detection. Moreover, Culotta *et al.* [14] proposed a method that extracts an individual's social network and then populates the address book of each member in the social network from the web using *conditional random fields*. Moreover, their method uses topics and the contact information to identify connected people who share the same information and build a social network.

Chen *et al.* [12] presented a general framework for crime data mining that uses several data mining techniques such as link analysis, association rule mining, and knowledge discovery to develop a crime-data-mining framework. The framework is able to identify crimes of different kinds. Chau *et al.* [9] improved efficiency and accuracy of a link analysis system by incorporating several techniques including co-occurrence analysis, the shortest path algorithm, and a heuristic approach to effectively identify connections in criminal networks. Techniques presented in the previous two studies have been applied to identify suspicious web communication, called *Dark Web*, for analyzing online

communication between suspects of the 9/11 attacks [10]. Xiang *et al.* [33] developed techniques for criminal information visualization that facilitate the decision making of investigators. Yang *et al.* [37] extracted criminal networks from online newswires, e.g., CNN.com, based on the similarity between stories related to different terrorist attacks and how these incidents evolved with the passage of time. Yang and Ng [36] further extended the technique to extract criminal networks from blog websites through a topic-based exploration mechanism. Xiong and Donath [34] displayed the specific attributes of the users involved in a chat and their relation within a chat room in a social network. The authors proposed a graphical representation: *portraits* of the attributes extracted from the chat contents represent the whole social environment including the users and their posting history. Moreover, Das and Das [51] proposed an unsupervised method to extract named entities and the relations among the extracted named entities in a crime report. They used noun phrase chunking to extract entities such as organizations, places, victims, and offenders involved in the crime, then applied a clustering technique to classify the entity pairs based on their context.

Many researchers analyzed structured data, e.g., police reports and newswire articles. They built the social networks using header information such as IP address and e-mail address. In the social network, graph, web pages, weblog, or email addresses are represented as nodes, and their relations are defined based on their communication measured in terms of inlink, outlink, and centrality [21]. In real life, even if the two entities never communicated with each other directly, they may be related. To identify this kind of relation, a content analysis of the conversation is essential. Similarly, an extracted social network based solely on keyword-similarity, not considering the semantics of the written text, could be misleading because in most malicious conversation the perceived meaning of written words is different from their apparent meaning, e.g., the word *"thunder"* means heroin, and *"snow"* means cocaine in drug trafficking communication. Concept Miner of our proposed framework is designed to identify the right context of a given word.

A probabilistic model called the *Author-Topic model* [28] is very effective in content categorization and has been employed by Zhou *et al.* [39] for discovering e-communities. Authors are represented by a probability distribution over topics, and each topic is denoted by a list of the most relevant words. Al-Zaidy *et al.* [4], [6] proposed a method to extract criminal communities from text documents taken from a confiscated hard drive. Their method was neither able to analyze the connections among the authors nor the semantics of the content, as we do in this paper. These two factors play a very significant role in crime investigation.

IV. PROBLEM STATEMENT

Suppose the forensic team or investigator has accessed the computer of a suspect S and found the archived chat log Φ from an IM system. Usually, a chat log consists of many chat sessions, and each chat session is further comprised of text messages exchanged between its participants. The main objective is to identify the communities (i.e., cliques) in the chat sessions, the relationships between the members of the communities, and the concepts that are discussed.

The problem is divided into two subproblems: *clique detection* and *concept analysis*.

A. SUBPROBLEM: CLIQUE DETECTION

The purpose of clique detection is to *efficiently* extract *all* the cliques from a given chat log. We discussed extensively with the digital forensic team of a Canadian law enforcement unit and formulated the following intuition of clique. In this paper, to make the discussion simple, we refer to a person's name as an entity.

If a group of entities frequently chat with each other, or if their names appear together in some minimum number of chat sessions, the group forms a clique.

In chat sessions, just counting the direct messages exchanged between two users is not enough to find all cliques. We propose that direct message exchange between all members of a clique is not necessary to consider those members in a clique. An entity ϵ is in a clique if 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.

Considering this generalized concept of a clique is very important, and it often reveals new indications for additional investigation. For instance, if two suspected entities ϵ_1 and ϵ_2 frequently chat with each other and mention the name of a third person ϵ_3 in the chat, there is a possibility that ϵ_3 is their "boss". Thus, ϵ_1, ϵ_2, and ϵ_3 form a clique even if ϵ_3 is not a user found in the chat log. This concept of clique may identify false positive cliques. For example, ϵ_3 mentioned by ϵ_1 and ϵ_2 is a celebrity. However, in crime investigation, the investigator will prefer spending more time classifying false positives and true positives rather than miss any potential evidence.

A chat log Φ is a set of chat sessions $\{\phi_1, \ldots, \phi_p\}$. Let $E(\Phi) = \{\epsilon_1, \ldots, \epsilon_u\}$ be set of all entities extracted from Φ. Let $E(\phi_i)$ be a set of entities extracted from chat session ϕ_i, where $E(\phi_i) \subseteq E(\Phi)$, for example, $E(\phi_5) = \{\epsilon_4, \epsilon_5, \epsilon_7\}$ in Table 1. Let $Y \subseteq E(\Phi)$ be a set of entities called *entityset*. A session ϕ_i contains an entityset Y if $Y \subseteq E(\phi_i)$. If there are total k entities in an entityset, it is called a *k-entityset*. For example, the entityset $Y = \{\epsilon_3, \epsilon_6, \epsilon_7\}$ is a 3-entityset. Percentage of chat sessions in Φ that contain entityset Y represents the support of Y. Only those entitysets become cliques whose support is equal or greater than the user defined support threshold.

Definition 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 min_sup is a real number in an interval of [0, 1]. A clique containing k entities is called a *k-clique*. ■

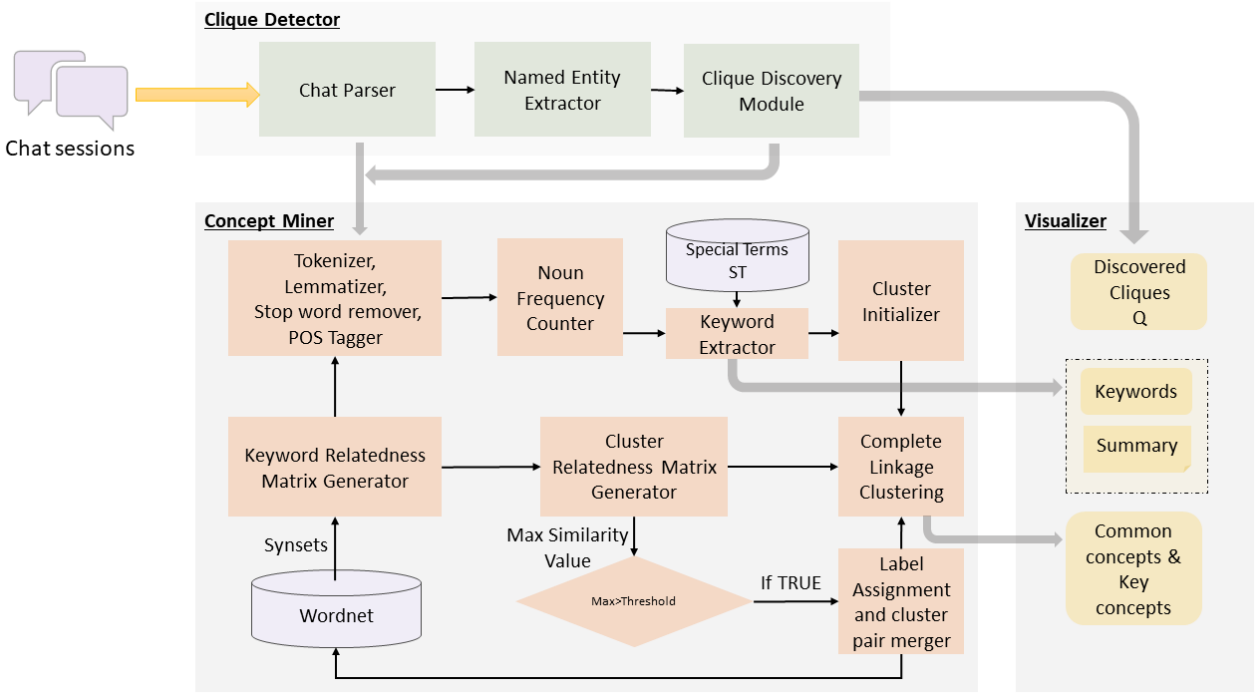

FIGURE 2. Proposed criminal information mining framework.

Definition 2 (Clique Detection): Let Φ be a collection of chat sessions. Let *min_sup* be a user-specified minimum support threshold. The subproblem of *clique detection* is to identify *all* cliques in Φ with respect to *min_sup*. ■

Example 3: Suppose the minimum support threshold value is 0.4. In Table 1, $\{\epsilon_4, \epsilon_5\}$ is not a clique because it has support 2/10 = 0.2. Similarly, $\{\epsilon_5, \epsilon_8\}$ is not a clique as it has support 3/10 = 0.3. $\{\epsilon_2, \epsilon_5\}$ is a 2-clique because its support is 4/10 = 0.4 and it contains 2 entities. ■

TABLE 1. Entities vector 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}	$\{\epsilon_1, \epsilon_5, \epsilon_7, \epsilon_8, \epsilon_9\}$

B. SUBPROBLEM: CONCEPT ANALYSIS

In some criminal cases, Canadian law enforcement officers have found thousands of chat users in the Windows Live Messenger chat log. This kind of large chat log may contain hundreds of cliques. The chat sessions of each discovered clique need to be further analyzed to filter the cliques involved in criminal activities. The objective of concept analysis is to extract the concepts and topics that highlight the semantics of the underlying chat conversations. A lexical database that captures the conceptual hierarchies of a language, e.g., WordNet [19], can be useful for concept analysis.

Definition 4 (Concept analysis): Let Q be a set of cliques discovered in Φ according to Definition 2. Let $\Phi(Q_i) \subseteq \Phi$ be the set of chat sessions contributing to the support of a clique $Q_i \in Q$. It is possible for a chat session to contribute to multiple cliques. Let H be a lexical database of the same language used in Φ. The purpose of *concept analysis* is to extract a set of key concepts, denoted by $KC(Q_i)$, for each discovered clique $Q_i \in Q$ using the lexical database H. The key concepts are the topics that bring the group of entities to form a clique. ■

V. PROPOSED APPROACH

The detailed architecture of our proposed framework is presented in Figure 2. It consists of three main components: *Clique Detector*, *Concept Miner*, and *Information Visualizer*. *Clique Detector* detects the cliques from the chat log. *Concept Miner* extracts the main concepts of the conversation from chat sessions of each identified clique. *Information Visualizer* allows the user to browse cliques and related concepts at multiple levels of abstraction in an interactive interface. The following sections describe each module in detail.

A. CLIQUE DETECTOR

Clique detection works in three main steps: dividing a chat log into sessions, entities extraction, and cliques detection.

1) CHAT LOG DIVISION INTO SESSIONS

A collection of messages exchanged between the users involved in conversation within a specific time period is called a session. In some IM systems, a session starts when the first message is sent between the participants of the conversation and ends when any of the participants closes the chat window. Once the chat window is closed, re-starting the chat is considered to be another session with a unique session ID in the log.

In some cases, the situation is more complex as it allows multiple users to chat simultaneously without any logical breakpoints for splitting the chat log into sessions. There are two solutions to this problem. The simple solution is to split the log into sessions based on some predefined unit of time, e.g., 15 minutes. The second and better solution is to calculate the time lag between consecutive messages, and when this lag passes a threshold time, consider it the start of a new session.

2) ENTITIES EXTRACTION

For entities extraction from each chat session, we used Stanford Named Entity Recognizer,[2] called *CRFClassifier*. The software provides a general implementation of linear chain Conditional Random Field (CRF) sequence model and is trained on the widely used named entity corpora. NER processes the chat session and extracts person name entities to identify the cliques in the chat session. This NER can be replaced with other NER tools if required, for instance, if there are non-English names in the chat log.

3) CLIQUES DETECTION

The clique detection operates on the entities extracted in the previous step. An entityset Y is a set of entities extracted from a chat log, and if its support is equal to or greater than a given threshold, it is considered as a clique. In case of a large number of identified entities $|E(\Phi)|$, it is very complex and time-consuming to generate all possible entitysets and calculate the support of each entityset in Φ, as there are $2^{|E(\Phi)|}$ possible combinations. To solve this problem, a modified Apriori algorithm [3] is used that extracts all cliques from Φ. The algorithm is described as follows:

Recall that $E(\Phi)$ represents the collection of entities in Φ, and $E(\phi_i)$ represents the collection of entities in a session $\phi_i \in \Phi$, where $E(\phi_i) \subseteq E(\Phi)$. The proposed *Clique Detector* is an iterative search algorithm that explores the $(k+1)$-cliques using the k-cliques. The generation of $(k+1)$-cliques from k-cliques is based on the following CM property.

Property 5 (CM property): All nonempty subsets of a clique are also cliques because $support(Y') \geq support(Y)$ if $Y' \subseteq Y$. ■

If $support(Y) < min\_sup$, then entityset Y is not a clique. According to the CM property, addition of an entity to a non-clique entityset cannot make the entityset a clique. Hence, if a k-entityset Y is not a clique, then generation of $(k+1)$-entityset $Y \cup \{\epsilon\}$ will be useless because $Y \cup \{\epsilon\}$ must not be a clique. $|\Phi(Y)|$ is the support of Y and is used to indicate the closeness among the entities in a clique Y.

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

Algorithm 1 Clique Detection

Input: Chat log Φ
Input: Threshold *min_sup*
Output: Cliques $Q = \{Q_1 \cup \cdots \cup Q_k\}$
Output: Chat sessions $\Phi(X), \forall X \in Q$

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

The proposed clique detection is summarized in Algorithm 1. The algorithm identifies the k-cliques from the $(k-1)$-cliques based on the CM property. At the initial step, Q_1 set is generated, which contains all 1-cliques X with $support(C_j) \geq min\_sup$. After generating the 1-cliques entityset, the next step is to generate the set of candidate 2-cliques, denoted by $Candidates_2$, using the set of 1-cliques from Q_1. Then it calculates the support of each candidate X in $Candidates_2$. All candidates in $Candidates_2$ that satisfy $|\Phi(X)| \geq min\_sup$ are 2-cliques, denoted by Q_2. The algorithm continues to generate Q_k from Q_{k-1} until $Candidate_k$ is empty.

The iteration through the data table and tracking of the chat session associated with each clique X in $Candidates_k$ is described in line 9-17. The algorithm iterates through the entities of each chat session $E(\phi)$ to find the match of each candidate entityset X. If it finds the match, the chat session ϕ is added to the set $\Phi(X)$. If the support $|\Phi(X)|$ is greater than or equal to the *min_sup*, then X is added to Q_k, the set of k-cliques with k members. When no further candidates can be generated, or when all candidate entitysets have support less than *min_sup* threshold, then the algorithm terminates and returns all cliques $Q = \{Q_1 \cup \cdots \cup Q_k\}$ with their associated chat sessions, except for the 1-cliques.

The complexity of Algorithm 1 is same as the complexity of the Apriori algorithm. Let c be the number of chat sessions, e be the number of distinct entities, and m be the *min_sup* threshhold. The generation of the set of size i and the support calculation of each set take $O(|e|^i)$ time and $O(|c|)$ time, respectively. Therefore, the time complexity is $O[(e+c)+(e^2+c)+(e^3+c)+\ldots] = O[m \times c + (e^1+e^2+\ldots+e^m)] = O(m \times c + \frac{(1-e^m)}{1-e})$. The efficient extraction of all frequent patterns is shown in the example below.

Example 6: Suppose, for the cliques identification from data in table 1, the value of *min_sup* = 0.4. The entities having support $\geq$ 0.4 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 $Q_2 = \{\{\epsilon_2, \epsilon_5\}, \{\epsilon_5, \epsilon_7\}\}$. Similarly, perform $Q_2 \bowtie Q_2$ to generate $Candidates_3 = \{\epsilon_2, \epsilon_5, \epsilon_7\}$ and determine $Q_3 = \emptyset$. Hence, the algorithm terminates and returns Q_2 and the associated chat sessions of every clique in Q_2. ■

B. CONCEPT MINER

Understanding the underlying semantics of the words is very important because criminals use different deception tactics to conduct their illicit activities. Due to unstructured and natural language data, understanding the true meaning and semantics is a difficult task. When malicious messages are read in a specific context, the symbols, abbreviations, and visual metaphors used in them reveal special meanings and evidence.

The objective of Concept Miner is to analyze and summarize the content of chat sessions into high-level concepts that can reveal some useful evidence. These high-level concepts are extracted from the set of associated chat sessions $\Phi(X)$ of every clique $X \in Q$ identified by Algorithm 1. Investigators can browse through these concepts to understand the semantics of the chat session. Concept Miner extracts *keywords*, *common concepts*, and *key concepts* from $\Phi(X)$. *Keywords* are frequent words, while *common concepts* are high-level concepts shared in the chat sessions. *Key concepts* are the top-ranked concepts by their importance. Concept Miner extracts the keywords from $\Phi(X)$ and groups them into clusters $\{\Omega_1, \cdots, \Omega_m\}$ by semantics. After clustering, it finds the common concepts CC of extracted keywords from each cluster Ω_i and, lastly, identifies the most important ones, the key concepts.

To extract the required information from the given chat logs, the following steps are required to be carried out.

1) TEXT PREPROCESSING

The following steps are required for preprocessing a text document.

1) Tokenization: Tokenization splits the text into different meaningful chunks of text such as words and phrases.
2) Stop word removal: In every document, there are some context-independent commonly used words that carry no semantic information. After tokenization, nondescriptive or irrelevant words that do not contribute to the understanding of the text are removed.
3) Lemmatization: lemmatization is a process of removing different inflectional forms of the word and grouping them together so that they can be analyzed as a single unit. Lemmatization performs morphological analysis with help of a dictionary to convert inflectional forms of a word into a base or dictionary form. This base form of a word is also called the lemma. Each lemma is considered for further processing.
4) POS tagging: part-of-speech tagging is performed to assign a lexical category to each word. In the proposed approach, only nouns are selected for further processing.

Following the preprocessing phase, there is a vector of terms connected to each chat session $\phi \in \Phi$ [26].

2) KEYWORD EXTRACTION

Frequent terms in a text document are considered important keywords [40]. Important keywords are extracted based on the frequency of their occurrence in the chat session. A threshold determines the minimum number of occurrences required per keyword to classify it as an important keyword. However, in crime investigation, some terms are very important even if they do not appear frequently. For example, some crime-related street terms such as ''cocaine'', ''marijuana'', or ''heroin'' require more attention. To extract such important keywords, the investigator can feed these special terms, denoted by ST, to the keyword extractor. In this implementation, we collected these terms from different law enforcement agencies and online sources.[3] A keyword is important in $\Phi(X)$ if it frequently appears in the chat session $\Phi(X)$ of clique $X \in Q$, but not in chat session $\Phi(Y)$ of another clique $Y \in Q$, where $X \neq Y$. $tf - idf$: term frequency−inverse document frequency is a statistical measure used to evaluate how important a word is to a document in a collection. To identify the important terms of each chat session, $tf - idf$ is calculated, and the top α of them are added to $KW(X)$, where α is a user-specified threshold. The sentences containing the important keywords can be used to generate a summary [38].

3) SEMANTIC-BASED CLUSTERING

The purpose of semantic-based clustering is to group the identified keywords into clusters$\{\Omega_1, \ldots, \Omega_m\}$ in such a way that the keywords in different clusters have low similarity while the keywords in the same cluster have high similarity.

We apply an agglomerative hierarchical clustering method to create the clusters [27]. In each step, it compares each pair of terms in $KW(X)$ and merges the most similar pair. $KW(X)$ is the set of important keywords extracted and can

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

Algorithm 2 Agglomerative Clustering

1. **INPUT:** Important Keywords $KW(X) \leftarrow \{KW_1, KW_2, KW_3, ...KW_n\}$
2. **INPUT:** Lexical Database DB
3. **INPUT:** Relatedness Measure RM
4. **INPUT:** Minimum Intra-cluster Similarity Value α
5. **OUTPUT:** List of Clusters $\Omega \leftarrow \{\Omega_1, \Omega_2, \Omega_3, ...\Omega_m\}$
6. Initialize Singleton Clusters
7. $\Omega \leftarrow GenerateSingletonClusters(KW)$
8. **while** $\Omega.Size \neq 1$ **do**
9. Relatedness Matrix M is an $s \times s$ matrix where $s = \Omega.Size$
10. $M \leftarrow GenerateRelatednessMatrix(\Omega, RM, DB)$
11. $MAX \leftarrow GetMaximumValueInMatrix(M)$
12. $MVC \leftarrow GetMaximumValueCoordinates(M)$
13. **if** $MVC.X \neq MVC.Y$ AND $MAX \geq \alpha$ **then**
14. $\Omega_x \leftarrow \Omega.get(MCV.X)$
15. $\Omega_y \leftarrow \Omega.get(MCV.Y)$
16. $\Omega.Remove(MCV.X)$
17. $\Omega.Remove(MCV.Y)$
18. $NSP \leftarrow FindNearestSynsetPairForClusters(\Omega_x, \Omega_y, RM, DB)$
19. **if** $NSP \neq NULL$ **then**
20. MergedCluster Ω_m
21. LeastCommonSubsumer $LCS \leftarrow FindLCS(NSP[0], NSP[1])$
22. ClusterLabel $CL \leftarrow SynsetToWord(LCS.GetSynset)$
23. CommonConcepts $CC \leftarrow$CL
24. $CC.AddChildren(\Omega_x.CC, \Omega_y.CC)$
25. $\Omega_m.Keyword \leftarrow Keyword(CC, LCS.GetSynset.GetPOS)$
26. $\Omega_m.AddChildren(\Omega_x)$
27. $\Omega_m.AddChildren(\Omega_y)$
28. $\Omega.Add(\Omega_m)$
29. **else**
30. *Continue*
31. **end if**
32. **else**
33. *Return* Ω
34. **end if**
35. **end while**
36. *Return* Ω

be represented as $KW(X) = \{KW_1, KW_2, KW_3, ...KW_n\}$, where n is total number of important keywords. Each KW_i represents an important keyword that contains the lemma, the POS tag, and the frequency of the important keyword in the context. Similarly, $\Omega(X)$ represents a list of clusters that can be represented as $\Omega = \{\Omega_1, \Omega_2, \Omega_3, ...\Omega_n\}$.

At the beginning of the process, each important keyword (i.e., KW_i) is placed in a cluster (i.e., Ω_i) of its own. We call it a cluster initialization step, which can be seen in line 7 of Algorithm 2. At this stage, the lemma of the keyword is considered as a label for each singleton cluster. After this step, the number of clusters is equal to the number of important keywords, and the merging process gets started. In subsequent phases, these clusters are combined sequentially until either all keywords end up being in a single cluster or some termination condition is reached. In our case, the termination condition is reached when the maximum value of similarity between all pairs of clusters is less than the given threshold α. α defines the maximum intra-cluster distance (minimum required similarity/relatedness) between members of the cluster. The higher the α, the more compact clusters are formed. At each step, two clusters with the shortest distance (i.e., maximum similarity/relatedness) are combined. The shortest distance between two clusters is estimated by following the complete-linkage clustering approach. In complete-linkage clustering, the distance of the link between two clusters is equal to the distance between those pairs of keywords (one in each cluster) that have a minimum value of similarity/relatedness (highest distance). This linkage method is also known as the farthest neighbor clustering. The resulting set of clusters will be considered as the compact conceptual representation of text. The complete process is depicted in Algorithm 2. The complexity of Algorithm 2 is equal to the complexity of the agglomerative clustering algorithm which is $O|n|^2$ where n is the number of important keywords.

4) RELATEDNESS MATRIX GENERATION

The two closest pair of clusters (which have the minimum distance between them) are identified by computing the similarity/relatedness between all pairs of clusters. Mathematically this process is represented as $n \times n$ size matrix. Suppose Ω denotes a n number of clusters list, then it is defined as $\Omega = \{\Omega_1, \Omega_2, \Omega_3, ...\Omega_n\}$, where each Ω_i contains one or more members (keywords). To compute the similarity/relatedness between all pairs of clusters, a matrix based computation is used as follows: Given a set of n clusters, M is an $n \times n$ relatedness matrix. $M_{i,j}$ denotes each cell of the matrix, where i and j represent the corresponding row and column of the matrix M. The relatedness value $R_{i,j}$ is computed for each pair of clusters representing the ith row and jth column of matrix M. The details of the computational steps required are given in Algorithm 3. The complexity of Algorithm 3 is $O|n|^2$ where n is the number of clusters which is very small.

5) SIMILARITY COMPUTATION BETWEEN PAIRS OF CLUSTERS

The minimum similarity (i.e., the least similar pair) among the pairs of keywords of two clusters is considered as a measure to define the closeness of pairs of clusters. As each cluster may represent a number of keywords, the similarity between two clusters is determined based on the minimum similarity between their members. This method of clustering is also known as complete-linkage clustering. Suppose Ω_1 and Ω_2 represents the pair of clusters whose relatedness has to be calculated. A denotes a set of keywords in

Algorithm 3 Relatedness Matrix Generation

INPUT: List of Clusters $\Omega \leftarrow \{\Omega_1, \Omega_2, \Omega_3, ...\Omega_n\}$
INPUT: Lexical Database *DB*
INPUT: Relatedness Measure *RM*
OUTPUT: Relatedness Matrix *M*
Relatedness Matrix M is an $n \times n$ matrix where $n = \Omega.Size$
for $i = 1$ to $\Omega.size$ **do**
  for $j = 1$ to $\Omega.size$ **do**
    if $i \neq j$ **then**
      R is a relatedness value between two farthest members of cluster Ω_i and Ω_j (one from each cluster)
      $R_{i,j} \leftarrow MinRelatedness(\Omega_i, \Omega_j, RM, DB)$
      $M_{i,j} \leftarrow R_{i,j}$
    else
      $M_{i,j} \leftarrow 0.0$
    end if
  end for
end for
Return M

cluster Ω_1, and B denotes the set of keywords in cluster Ω_2. Following complete-linkage clustering, our goal is to find the minimum relatedness/similarity between the pair of keywords (one from each cluster). Mathematically this calculation can be represented by the matrix M of size $a \times b$, where a and b denote the number of keywords in cluster A and B. $M_{i,j}$ represents the value of relatedness/similarity of the keywords representing the ith row and jth column of the matrix. The minimum relatedness value *MRV* holds the value of relatedness between the two least related/similar pairs of keywords. Algorithm 4 lists all the steps required to perform the computation.

Algorithm 4 Relatedness/Similarity Computation Between Pair of Clusters

INPUT: Cluster Ω_1
INPUT: Cluster Ω_2
INPUT: Relatedness Measure *RM*
INPUT: Lexical Database *DB*
OUTPUT: Minimum Relatedness Value *MRV*
$A \leftarrow$ List of keywords in cluster Ω_1
$B \leftarrow$ List of keywords in cluster Ω_2
$M \leftarrow GetKeywordsBasedRelatednessMatrix(A, B, RC, DB)$
$MRV \leftarrow GetMinValueInMatrix(M)$
Return *MRV*

6) SIMILARITY COMPUTATION BETWEEN PAIR OF KEYWORDS

In order to compute the semantic relatedness/similarity between a pair of keywords, we first need to find the true sense of each keyword. This process is known as Word Sense Disambiguation (WSD), i.e., to fix the ambiguous terms by finding their true meaning. As explained earlier, our strategy for WSD is based on the distributional hypothesis. Therefore, only those senses of keywords are considered as true senses that demonstrate the maximum level of similarity against the context.

Suppose KW_1 and KW_2 are a pair of keywords. $LS1$ and $LS2$ are the lists of all possible synsets for keywords KW_1 and KW_2, respectively. A lexical database (WordNet in our case) provides the list of all possible synsets for each keyword. To compute relatedness value RV between KW_1 and KW_2, each pair of synsets (one from each keyword) is compared (using suitable semantic relatedness measure RM) to measure the degree of semantic relatedness. This is also represented by matrix computation of $x \times y$ matrix. The synset pair that demonstrates the highest value for relatedness (represented as *maximumScore*) is regarded as holding true senses of keywords KW_1 and KW_2, respectively. Algorithm 5 provides the sequence of steps to achieve this computation.

Algorithm 5 Relatedness Calculator for Keywords

INPUT: Keyword KW_1 , KW_2
INPUT: Relatedness Measure *RM*
INPUT: Lexical Database *DB*
OUTPUT: Relatedness Value *RV*
Maximum Score *MaximumScore*
POS Value $PV1 \leftarrow KW_1.getPosValue()$
POS Value $PV2 \leftarrow KW_2.getPosValue()$
Lemma Value $LV1 \leftarrow KW_1.getLemma()$
Lemma Value $LV2 \leftarrow KW_2.getLemma()$
List of Synsets $LS1 \leftarrow DB.getAllSynsets(LV1, PV1)$
List of Synset $LS2 \leftarrow DB.getAllSynsets(LV2, PV2)$
for $i = 1$ to $LS1.size$ **do**
  for $j = 1$ to $LS2.size$ **do**
    Relatedness $R \leftarrow RM.getRelatednessOfSynsets(LS1_i, LS2_j)$
    Score $Score \leftarrow R.getScore()$
    if $Score > MaximumScore$ **then**
      $MaximumScore \leftarrow Score$
    end if
  end for
end for
$RV \leftarrow MaximumScore$
Return *RV*

7) CLUSTER LABELING

When two clusters are merged into one cluster, there is a need to label the resulting merged cluster. This label is viewed as the common concept *CC* or subject of the new cluster. It represents all the keywords/members in the subject cluster. During the initialization phase, the important keyword (i.e., the only member of the singleton cluster) is used as a label term of each singleton cluster. During subsequent phases, the LCS (least common subsumer) is obtained by comparing the two nearest synset nodes that represent the topic of each

cluster being compared. For example, if clusters Ω_1 and Ω_2 are compared, their topics as T_1 and T_2 are used as keywords. These keywords are then evaluated to find all the synsets of CC_1 and CC_2. The closest synsets pair is obtained and their LCS is used as the label of the newly merged cluster. The LCS is obtained from the shortest path that connects two synsets in the WordNet hierarchy. It is important to note that the topics CC_1 and CC_2 are the keywords that contain the lexical term with its POS tag. For example in Table 2, the word *"Thunder"* has three senses (synsets). In the context of drug smuggling, *"Thunder"* means *"heroin"*, but it can have a different meaning in other contexts. The sense is selected according to the context described by other terms in the same cluster, which is described below. The objective is to find the common hypernym/Least common subsumer of the keywords of both merged clusters. The LCS obtained is the synset (node in WordNet ontology), so the selection of the proper word (lexical term) among other members needs to be resolved. Currently, the first member is selected (as default). This is how each cluster is assigned a common concept, and then the common concept term is afterward compared with another common concept term. The comparison of two common concept terms is the calculation of the similarity measure between two nearest synsets of words. Then the LCS of these two synsets is the common concept of the resulting merged cluster. Each cluster also maintains a data structure of type tree and stores all the cluster labels as common concepts *CC*. Investigators can drill down and roll up in this tree in the Information Visualizer window according to their requirements.

According to the evaluation, the identified common concepts truly represent the semantic of the chat sessions connected with a clique; however, in real applications, the number of common concepts is very large, and it is not feasible to show *all* of them in the visualizer. To solve this issue, the common concepts were ranked, and the top β of them were displayed, where β is a user-specified threshold. Common concepts may have different importance in a crime context, which is measured by calculating the sum of the $tf - idf$ values of the matched terms normalized by a total number of common concepts.

8) EXTRACTING MISCELLANEOUS INFORMATION

This step extracts relevant information such as phone numbers, addresses, e-mails, or website URLs from the chat sessions of every clique. This task can be easily accomplished by using regular expressions.

TABLE 2. ***Synsets*** **and** ***direct hypernyms*** **of selected terms retrieved from WordNet.**

Term	Synsets → Direct Hypernyms
Thunder	1. boom, roar, roaring, thunder (a deep prolonged loud noise) → noise (sound of any kind (especially unintelligible or dissonant sound)) 2. thunder (a booming or crashing noise caused by air expanding along the path of a bolt of lightning) → noise (sound of any kind (especially unintelligible or dissonant sound)) 3. big H, hell dust, nose drops, smack, thunder, skag, scag (street names for heroin) → heroin, diacetylmorphine (a narcotic that is considered a hard drug; a highly addictive morphine derivative; intravenous injection provides the fastest and most intense rush)
Smack	1. slap, smack (a blow from a flat object (as an open hand)) → blow, bump (an impact (as from a collision)) 2. relish, flavor, flavour, sapidity, savor, savour, smack, nip, tang (the taste experience when a savoury condiment is taken into the mouth) → taste, taste sensation, gustatory sensation, taste perception, gustatory perception (the sensation that results when taste buds in the tongue and throat convey information about the chemical composition of a soluble stimulus) 3. smack (a sailing ship (usually rigged like a sloop or cutter) used in fishing and sailing along the coast) → sailing vessel, sailing ship (a vessel that is powered by the wind; often having several masts)
Nose drops	1. big H, hell dust, nose drops, smack, thunder, skag, scag (street names for heroin) → heroin, diacetylmorphine (a narcotic that is considered a hard drug; a highly addictive morphine derivative; intravenous injection provides the fastest and most intense rush)
Heroin	1. heroin, diacetylmorphine (a narcotic that is considered a hard drug; a highly addictive morphine derivative; intravenous injection provides the fastest and most intense rush) → opiate (a narcotic drug that contains opium or an opium derivative) → hard drug (a narcotic that is considered relatively strong and likely to cause addiction)

C. INFORMATION VISUALIZER

Information Visualizer provides an interactive user interface to browse the extracted information such as cliques, concepts, and terms. Information Visualizer displays the clique as a graph whose nodes represent the entities, the edges represent the relationship between entities, and the closeness of entities is shown by the lengths of edges. In case of a large number of discovered cliques, visualization becomes very challenging. Recall that Property 5 states that every subset of a clique is also a clique, so the discovered cliques represent multiple layers of relationships. The data associated with each clique contains closeness, keywords, common concepts, key concepts, and other relevant information. A tool called prefuse [1] is used to design an intuitive interface that allows the user to view a clique at higher and lower levels of abstraction.

VI. EXPERIMENTS AND DISCUSSION

Experimental evaluation has three objectives: The first objective is to evaluate the Clique Detector to see if the extracted cliques represent meaningful communities of individuals in the real world and to find the effect of the minimum support threshold on the number of cliques. The second objective is to evaluate if the Concept Miner can precisely extract the semantics of the conversation of each extracted clique. The third objective is to measure the efficiency and scalability of the presented framework in terms of the impact of the input parameter, execution time, and data size.

A. DATASET

For clique mining, we use two types of dataset. One is a real-life dataset on a hard drive confiscated from the computer of

a hacker, provided by Sûreté du Québec (SQ), the Québec provincial police. Due to confidentiality and privacy issues, some information is masked. The second dataset is created synthetically by our researchers using the information collected from an anonymous source. The dataset is available for download at our website.[4] For evaluation of concept mining, we use the Reuters-21578 [41] dataset, which is the most widely used test collection for categorization research and contains a large collection of manually annotated newswire stories. Each story is annotated with one or more topics assigned manually. A sample of 44 stories is randomly selected from the dataset, where each story is considered as a single document.

B. CLIQUE DETECTOR

This section discusses our experiments for clique detection for two different datasets.

1) REAL LIFE CRIMINAL CASE

To evaluate the effectiveness of our proposed method in a real-life criminal case, we conduct experiments on a hard disk provided by the Sûreté du Québec (SQ). The hard disk was confiscated from a suspect who had allegedly been involved in conducting cyber crimes in 2005. Though the matter has been investigated and the criminal prosecuted, we reinvestigate the case using our method to gauge its effectiveness compared with that of a human expert. No prior knowledge about the cybercrime is given to us. We install our framework on a workstation connected to the suspect's hard disk. The analysis is semi-automatic because the user needs to input and adjust certain parameters, such as the minimum support threshold during the process.

The result of the analysis is depicted in Figure 3, which displays the discovered cliques associated with the suspect. The two central nodes in the figure, denoted by *SA1* and *SA2*, represent two chat log accounts owned by the same suspect. Our proposed method successfully identifies a suspect's suspicious chat conversation with an entity labeled *Potential Collaborator*. The conversation contains a frequent term *botnet*. The identified personal information includes e-mail addresses, URLs, IP addresses, and phone numbers. Some of the information is masked in the figure due to confidentiality.

By carefully examining the detailed output of the result for this particular clique, as shown in Figure 4, the following findings could be drawn. There are several e-mail addresses in a format similar to "abc999@173.206.170.208.dsl.some_isp.ca". It is clear that the e-mail address is comprised of a prefix of random letters or numbers followed by the "@" symbol. An IP address, followed by terms such as "dsl" or "modem", ends with the domain name of some Internet Service Provider (ISP). Examining the last part of the e-mail address "173.206.170.208.dsl.some_isp.ca", an investigator with an understanding of ISP networks will know that it is a hostname that the DNS server of a service provider will have in its database for a client that receives a temporary dynamic IP address from the DHCP server. It is highly unusual to have an e-mail server running on a temporary hostname and IP address that changes dynamically. Therefore, these are not real e-mail servers. Instead, these are some of the bots controlled by the suspect via port 25 (SMTP port). Moreover, some website URLs are in the form similar

[4]http://www.ciise.concordia.ca/~fung/research/data/MSNlogs.zip

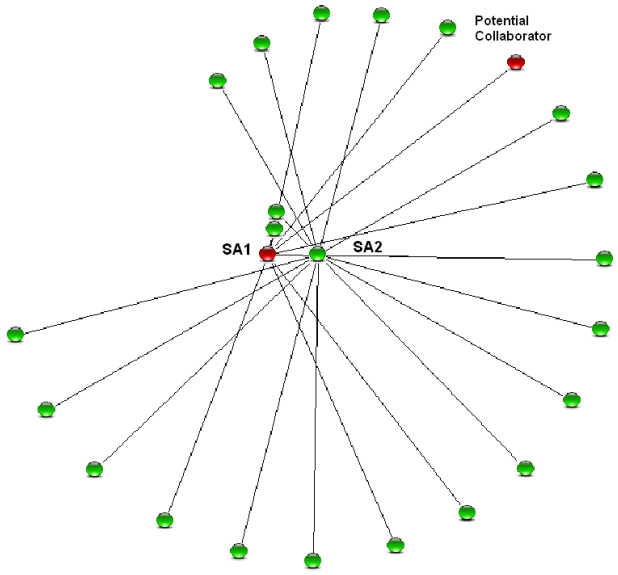

FIGURE 3. Result of the Clique Detector for the confiscated hard disk (due to privacy concerns, names and other identifying information are omitted).

FIGURE 4. Forensic analysis result of the confiscated hard disk.

to ''http://yyyyyyy.free.com/save.exe''. Here the URLs seem to contain the domain name of a free Web hosting company, whereas the subdomain is the user account on the hosting company. The URLs are pointing to a binary executable file named ''save.exe'' or ''sms.exe''. An experienced investigator can conclude that these are most likely the URLs to the binary executables of the bot that the suspect uses to spread the malware to victims.

Our developed software and analysis results have been evaluated by two cybercrime investigators at SQ. They agreed that our software can serve as an important tool in their investigation process, especially in the initial phase when they do not have enough clues to start with. Our developed framework can be applied to most online documents, although the current study focuses on analyzing chat sessions. The chat sessions used in our experiments were collected from MSN and Windows Live Messenger, but the framework can easily be extended to support other chatting tools.

2) CHAT LOG

A screenshot of our developed framework in Figure 5 shows the chat log of extracted cliques. There are ten extracted cliques in the figure, where each clique contains two or three entities. All of the extracted cliques contain the central node represent the suspect, while the other nodes represent the entities linked with the suspect. The relationship between the entities is indicated by edges connecting the entities. Using this visualizer, the investigator can select a clique in order to display all relevant information about it. For example, the clique containing entities *BANG54321033*, *EDDYSPHARMACY8*, and *CHARLIE* is important as the drug-related terms, e.g., grass, heroin, and snow, are found in the chat conversation of its members. The identified entities and the cliques are manually compared with the content of the chat sessions. Results show that the system is able to extract 80% of the cliques correctly, with some false positive cases. Figure 6 shows the number of extracted cliques with respect to the minimum support threshold. Change of minimum support threshold from 0.33% to 3.33% results in quickly decreasing the number of cliques from 155 to 8.

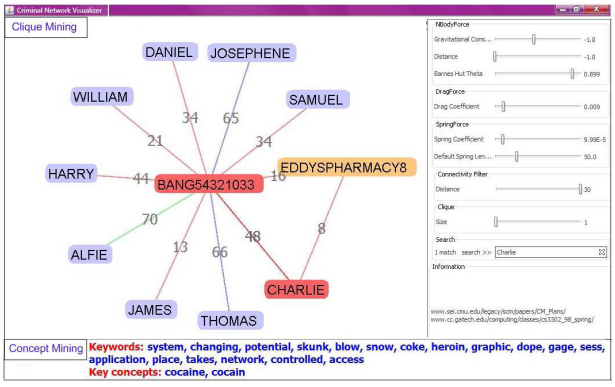

FIGURE 5. A screenshot of the developed framework.

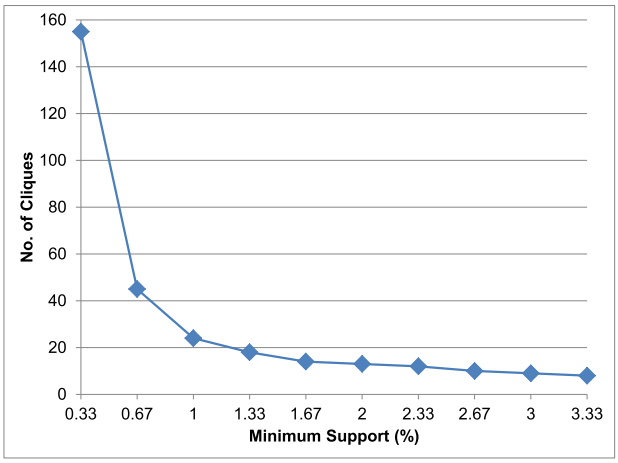

FIGURE 6. Impact of minimum support on the number of cliques.

C. CONCEPT MINER

We also evaluate the concept analysis functionality of the presented framework and compare it with state-of-the-art clustering algorithms. The evaluation is divided into two phases. The first phase compares the performance of our clustering algorithm in terms of different semantic similarity measures (i.e., Jiang & Conrath; Resnik; Leakcock & Chodorow; Wu & Palmer; Lin; Lesk). In the second phase, we perform an experimental evaluation of the proposed system by comparing it with state-of-the-art document clustering methods, agglomerative UPGMA [42], [43], k-means [42], and bisecting k-means [42]–[44].

A common evaluation method, F-measure [44], [45] is employed to measure the accuracy of the produced clustering solutions. It is the harmonic mean of recall and precision and can be calculated as shown in Equation 1. It is considered to be the standard evaluation method for both flat and hierarchical clustering solutions. Hatzivassiloglou and McKeown (1993) presented a pair-wise precision and recall method to evaluate the results produced by clustering algorithms specific to NLP [46].

Given an element set $E = \{E_1, E_2, E_3, ..., E_n\}$ with n elements, $\Omega = \{\Omega_1, \Omega_2, \Omega_3, ..., \Omega_k\}$ represents the set of elements produced by clustering solutions, and $M = \{M_1, M_2, M_3, ..., M_l\}$ represents the set of elements that are manually classified as gold standard. Ω and M represent the two partitions of element set E. Ω_i denotes the set of elements in the ith cluster of partition Ω, and M_j denotes the set of elements in the jth cluster of partition M. The number of common pairs in M and Ω are true positive tp values, false positive fp values are the number of pairs in Ω, but not in M, and false negative fn values are the number of pairs in M, but not in Ω. This can be represented in Equations 2 and 3 for precision and recall.

$$F - measure = \frac{2 \times recall \times precision}{recall + precision} \quad (1)$$

$$recall = \frac{tp}{fn + tp} \tag{2}$$

$$precision = \frac{tp}{fp + tp} \tag{3}$$

In the first phase of our evaluation, the proposed technique is applied to each document, thus producing a compact conceptual representation of the document in the form of concepts and relationships among them. Performing agglomerative hierarchical clustering produces a cluster of concepts with strong cohesive relationships. The α value determines the compactness of the clusters formed; the higher the α value, the more compact the clusters are formed. To determine the optimum value of α for each semantic similarity/relatedness measure, we performed several experiments on the Reuters-21578 dataset and recorded the best α value for each measure. These values are shown in Table 3. The label assigned to the resulting clusters represents the common concepts of the document. The label of the cluster is matched with the topic of the newswire story, and results are recorded in terms of true positive, false positive, and false negative. The results are shown in Figure 7.

TABLE 3. Intra-cluster similarity threshold.

Similarity Measure	α
J&C	0.15
Lin	0.60
W&P	0.80
Lesk	1
L&C	0.14
Resnik	0.35

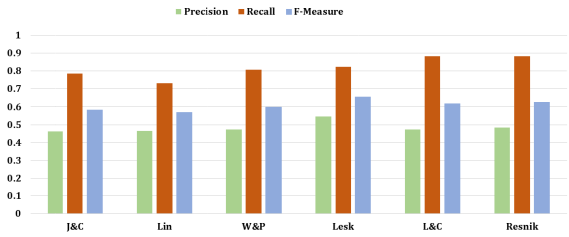

FIGURE 7. Comparisons of six popular semantic similarity measures on Reuters dataset.

It is observed that Lesk F-measure is the highest, while Resnik and L&C are close. There are fewer false negatives identified and recall remained higher. This is due to the fact that the Reuters dataset came under the category of formal conversation based datasets. Each newswire story contained normal English sentences, and no slang or informal conversation occurred.

In the second phase of our evaluation, we compare our clustering algorithm with other popular clustering algorithms. We use Cluto version 2.1.2 [47] to generate the results of agglomerative UPGMA, k-means, and bisecting k-means clustering methods. The produced clustering results are then evaluated to calculate the F-measure. The comparison is shown in Figure 8. As we can see, our proposed clustering solution outperforms other popular clustering algorithms by a significant margin. It also produces better recall and F-measure scores in comparison to its competitors.

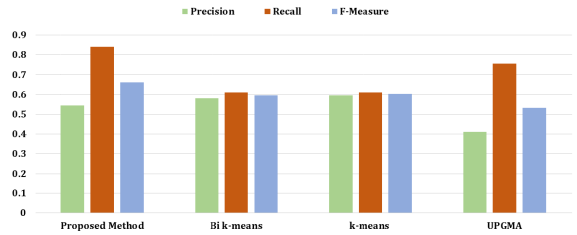

FIGURE 8. Comparison with other clustering algorithms on Reuters.

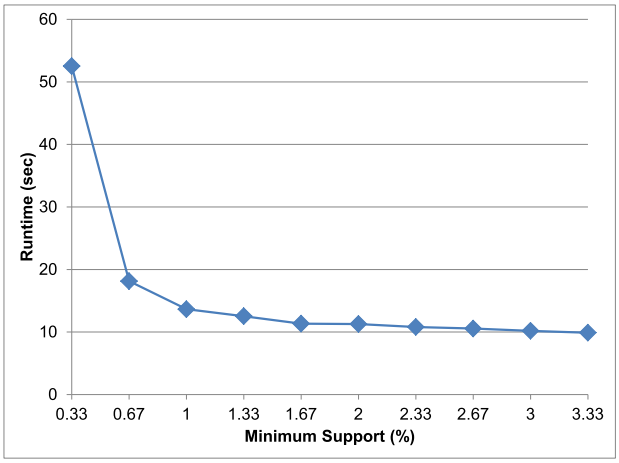

FIGURE 9. Efficiency [Execution time vs. Minimum support].

The Concept Miner extracts the keywords, common concepts, and key concepts from the chat session of each identified clique. The information extracted by Concept Miner associated with each clique is shown in Figure 5. The framework provides different levels of abstraction to browse the cliques and the summary of their conversation.

The results of Concept Miner for the chat log of *BANG54321033*, *EDDYSPHARMACY8*, and *CHARLIE* are interesting. The extracted keywords, such as blow, snow, coke, dope, and gage, are street terms used to represent cocaine, a narcotic. The words "cocaine" and "cocain", identified as the key concepts, represent the main topic of chat conversation of the aforementioned clique. Hence by comparing the results, it is proved that topics identified by Concept Miner can truly represent the semantics of the chat log.

Furthermore, we evaluate the efficiency and the scalability of the framework. In Figure 9, the runtime is plotted against the minimum support to measure the efficiency of the developed framework. As the minimum support increases from 0.33% to 3.33%, it decreases the number of cliques. Hence, the total runtime decreases from 53 seconds to 10 seconds. To measure the scalability, Figure 10 plots the runtime of different phases with respect to the data size from 1,000 sessions to 10,000 sessions, with a minimum support threshold

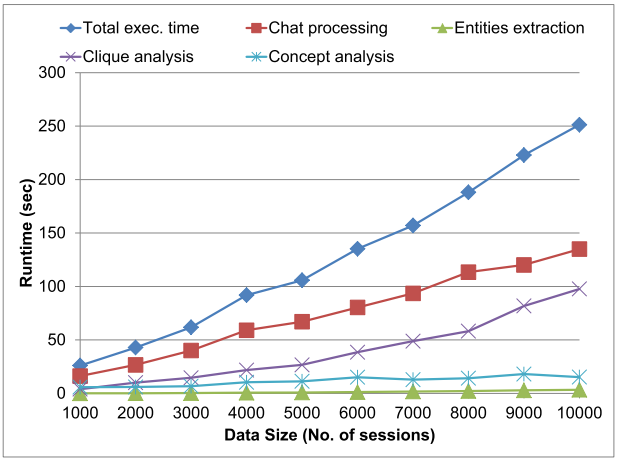

FIGURE 10. Scalability [Execution time vs. Data size].

fixed at 0.67%. The calculated execution time suggests the scalability of the proposed framework as runtime increases linearly with the increase in data size.

VII. CONCLUSION

In this paper, we present a WordNet-based criminal information mining framework to identify and extract forensically relevant information from large suspicious chat logs. The framework processes the chat log of a suspect to identify a set of cliques and topics in the conversation of each clique. The results of the experimental evaluation suggest that the proposed framework can efficiently extract the cliques and the semantics of the conversation between members of the cliques. Results also demonstrate the efficiency and scalability of the method. By using a more accurate similarity measure, we compare the accuracy of topic classification of our system with other state-of-the-art clustering algorithms and prove that our system produced 10% to 20% more accurate clusters and their topics. We closely collaborated with a cyber forensics team in Canada during the design and development phase of the framework. The effectiveness of the proposed framework is confirmed by experienced crime investigators.

The topics obtained from clustering results could be matched with crime ontologies to find out whether or not the suspect is involved in any suspicious activities. However, this requires further research in the area. We aim to address this topic in our future research.

ACKNOWLEDGMENT

In addition, the authors would like to thank Mr. Mirza Ahmed for his contributions in the early stage of the project.

REFERENCES

[1] *Prefuse: Information Visualization Toolkit*. Accessed: Sep. 1, 2010. [Online]. Available: http://prefuse.org/download/

[2] A. Abbasi and H. Chen, "Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace," *ACM Trans. Inf. Syst.*, vol. 26, no. 2, pp. 1–29, Mar. 2008.

[3] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Washington, DC, USA, Jun. 1993, pp. 207–216.

[4] R. Al-Zaidy, B. Fung, and A. M. Youssef, "Towards discovering criminal communities from textual data," in *Proc. ACM Symp. Appl. Comput.*, Taichung, Taiwan, Mar. 2011, pp. 172–177.

[5] E. Alfonseca and S. Manandhar, "An unsupervised method for general named entity recognition and automated concept discovery," in *Proc. Int. Conf. General WordNet*, 2002, pp. 34–43.

[6] R. Al-Zaidy, B. C. M. Fung, A. M. Youssef, and F. Fortin, "Mining criminal networks from unstructured text documents," *Digit. Invest.*, vol. 8, nos. 3–4, pp. 147–160, Feb. 2012.

[7] M.-H. Antoni-Lay, G. Francopoulo, and L. Zaysser, "A generic model for reusable lexicons: The genelex project," *Literary Linguistic Comput.*, vol. 9, no. 1, pp. 47–54, 1994.

[8] V. R. Carvalho and W. W. Cohen, "Learning to extract signature and reply lines from email," in *Proc. Conf. Email Anti-Spam*, Mountain View, CA, USA, Jul. 2004.

[9] J. Schroeder, J. Xu, H. Chen, and M. Chau, "Automated criminal link analysis based on domain knowledge," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 58, no. 6, pp. 842–855, Apr. 2007.

[10] H. Chen, W. Chung, J. Qin, E. Reid, M. Sageman, and G. Weimann, "Uncovering the dark Web: A case study of Jihad on the Web," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 59, no. 8, pp. 1347–1359, Jun. 2008.

[11] H. Chen *et al.*, "Crime data mining: An overview and case studies," in *Proc. Annu. Nat. Conf. Digit. Government Res.*, Boston, MA, USA, May 2003, pp. 1–5.

[12] 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, pp. 50–56, Apr. 2004.

[13] N. Chinchor, "Overview of MUC-7," in *Proc. 7th Message Understand. Conf.*, Fairfax, VA, USA, May 1998.

[14] A. Culotta, R. Bekkerman, and A. Mccallum, "Extracting social networks and contact information from email and the Web," in *Proc. 1st Conf. Email Anti-Spam*, Mountain View, CA, USA, Jan. 2004, pp. 1–8.

[15] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics*, Stroudsburg, PA, USA, Jun. 2005, pp. 363–370.

[16] T. Kucukyilmaz, B. B. Cambazoglu, C. Aykanat, and F. Can, "Chat mining: Predicting user and message attributes in computer-mediated communication," *Inf. Process. Manage.*, vol. 44, no. 4, pp. 1448–1466, 2008.

[17] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th Int. Conf. Mach. Learn.*, San Francisco, CA, USA, 2001, pp. 282–289.

[18] J. S. McIllwain, "Organized crime: A social network approach," *Crime, Law Social Change*, vol. 32, no. 32, pp. 301–323, Dec. 1999.

[19] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995.

[20] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticæ Investigationes*, vol. 30, no. 1, pp. 3–26, Jan. 2007.

[21] M. E. J. Newman, S. Forrest, and J. Balthrop, "Email networks and the spread of computer viruses," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 66, no. 3, Sep. 2002, Art. no. 035101.

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

[23] T. Pedersen, S. Patwardhan, and J. Michelizzi, "WordNet::Similarity: Measuring the relatedness of concepts," in *Proc. HLT-NAACL*, Boston, MA, USA, May 2004, pp. 1024–1025.

[24] N. Pendar, "Toward spotting the pedophile telling victim from predator in text chats," in *Proc. Int. Conf. Semantic Comput.*, Sep. 2007, pp. 235–241.

[25] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[26] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Boston, MA, USA: Addison-Wesley, 1989.

[27] D. K. Sepandar, D. Klein, and D. M. Christopher, "Interpreting and extending classical agglomerative clustering algorithms using a model-based approach," in *Proc. 19th Int. Conf.*, Sydney, NSW, Australia, Feb. 2002, pp. 283–290.

[28] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths, "Probabilistic author-topic models for information discovery," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Seattle, WA, USA, Aug. 2004, pp. 306–315.
[29] S. J. Stolfo and S. Hershkop, "Email mining toolkit supporting law enforcement forensic analyses," in *Proc. Nat. Conf. Digit. Government Res.*, May 2005, pp. 221–222.
[30] G.-F. Teng, M.-S. Lai, J.-B. Ma, and Y. Li, "E-mail authorship mining based on SVM for computer forensic," in *Proc. 3rd Int. Conf. Mach. Learn. Cybern.*, Aug. 2004, pp. 1204–1207.
[31] G. Wang, H. Chen, and H. Atabakhsh, "Automatically detecting deceptive criminal identities," *Commun. ACM*, vol. 47, no. 3, pp. 70–76, Mar. 2004.
[32] J. Wolak, K. Mitchell, and D. Finkelhor, "Online victimization of youth: Five years later," Nat. Center Missing Exploited Children Bull., Alexandria, VA, USA, Tech. Rep. 07-06-025, 2006.
[33] Y. Xiang, M. Chau, H. Atabakhsh, and H. Chen, "Visualizing criminal relationships: Comparison of a hyperbolic tree and a hierarchical list," *Decis. Support Syst.*, vol. 41, no. 1, pp. 69–83, Nov. 2005.
[34] R. Xiong and J. Donath, "PeopleGarden: Creating data portraits for users," in *Proc. 12th Annu. ACM Symp. User Interface Softw. Technol.*, New York, NY, USA, Nov. 1999, pp. 37–44.
[35] C. C. Yang, N. Liu, and M. Sageman, "Analyzing the terrorist social networks with visualization tools," in *Proc. Intell. Secur. Inform.*, San Diego, CA, USA, 2006, pp. 331–342.
[36] C. C. Yang and T. D. Ng, "Terrorism and crime related Weblog social network: Link, content analysis and information visualization," in *Proc. IEEE Intell. Secur. Inform.*, New Brunswick, NJ, USA, May 2007, pp. 55–58.
[37] C. C. Yang, X. Shi, and C.-P. Wei, "Tracing the event evolution of terror attacks from on-line news," in *Proc. IEEE Int. Conf. Intell. Secur. Inform.*, San Diego, CA, USA, 2006, pp. 343–354.
[38] V. A. Yatsko and T. N. Vishnyakov, "A method for evaluating modern systems of automatic text summarization," *Autom. Document Math. Linguistics*, vol. 41, no. 3, pp. 93–103, Jun. 2007.
[39] D. Zhou, R. Manavoglu, J. Li, C. L. Giles, and H. Zha, "Probabilistic models for discovering e-communities," in *Proc. 15th Int. Conf. World Wide Web*, New York, NY, USA, May 2006, pp. 173–182.
[40] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.
[41] D. Lewis. (1997). *Reuters-21578 Dataset*. [Online]. Available: http://www.daviddlewis.com/resources/testcollections/reuters21578/
[42] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, 1988.
[43] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ, USA: Wiley, 1990.
[44] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering technique," in *Proc. KDD Workshop Text Mining*, Aug. 2000.
[45] B. Larsen and C. Aone, "Fast and effective text mining using linear-time document clustering," in *Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 1999, pp. 16–22.
[46] V. Hatzivassiloglou and R. K. McKeown, "Predicting the semantic orientation of adjectives," in *Proc. 35th Annu. Meeting Assoc. Comput. Linguistics*, Stroudsburg, PA, USA, Jul. 1997, pp. 174–181.
[47] G. Karypis, "CLUTO—A clustering toolkit," Dept. Comput. Sci. Eng., Univ. Minnesota, Minneapolis, MN, USA, Tech. Rep. 02-017, Nov. 2003.
[48] T. Y. Yang, A. Dehghantanha, K.-K. R. Choo, and Z. Muda, "Investigating america online instant messaging application: Data remnants on windows 8.1 client machine," in *Proc. Contemp. Digit. Forensic Invest. Cloud Mobile Appl.* Amsterdam, The Netherlands: Elsevier, 2017, pp. 21–39.
[49] P. Magalingam, S. Davis, and A. Rao, "Using shortest path to discover criminal community," *Digit. Invest.*, vol. 15, pp. 1–17, Dec. 2015.
[50] I. Awan, "Cyber-extremism: Isis and the power of social media," *Society*, vol. 54, no. 2, pp. 138–149, Apr. 2017.
[51] P. Das and A. K. Das, "Crime pattern analysis by identifying named entities and relation among entities," in *Advanced Computational and Communication Paradigms*. Singapore: Springer, 2018, pp. 75–84.
[52] H. A. Shabat and N. Omar, "Named entity recognition in crime news documents using classifiers combination," *Middle-East J. Sci. Res.*, vol. 23, no. 6, pp. 1215–1221, 2015.
[53] F. Iqbal, B. C. M. Fung, and M. Debbabi, "Mining criminal networks from chat log," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, vol. 1, Dec. 2012, pp. 332–337.

FARKHUND IQBAL received the master's and Ph.D. degrees from Concordia University, Canada, in 2005 and 2011, respectively.

He is currently an Associate Professor and the Director of the Advanced Cyber Forensics Research Laboratory, College of Technological Innovation, Zayed University, United Arab Emirates. He is also an Affiliate Professor with the School of Information Studies, McGill University, Canada, and an Adjunct Professor with the Faculty of Business and IT, University of Ontario Institute of Technology, Canada. He is using machine learning and big data techniques for problem solving in healthcare, cybersecurity, and cybercrime investigation in smart and safe city domain. He has published more than 80 papers in peer-reviewed high-ranked journals and conferences.

Dr. Iqbal is a member of several professional organizations, including the ACM and IEEE Digital Societies. He was a recipient of several prestigious awards and research grants. He has served as the Chair and a TPC Member of several IEEE/ACM conferences, a Guest Editor for special issues, and a Reviewer for high-rank journals.

BENJAMIN C. M. FUNG received the Ph.D. degree in computing science from Simon Fraser University, in 2007.

He is currently the Canada Research Chair in data mining for cybersecurity and an Associate Professor with the School of Information Studies, McGill University. He has over 120 refereed publications that span the research forums of data mining, machine learning, privacy protection, cyber forensics, and building engineering. His data mining works in crime investigation and authorship analysis have been reported by Media Worldwide.

Dr. Fung is a licensed Professional Engineer in software engineering. He is a Senior Member of the ACM and the IEEE. He is a Co-Curator of cybersecurity in the World Economic Forum.

MOURAD DEBBABI received the M.Sc. and Ph.D. degrees in computer science from Paris-XI Orsay, University, France.

He was the Specification Lead of four Standard Java Intelligent Networks Java Specification Requests dedicated to the elaboration of standard specifications for presence and instant messaging. He is currently a Full Professor with the Concordia Institute for Information Systems Engineering and an Associate Dean for Research and Graduate Studies with the Faculty of Engineering and Computer Science, Concordia University. He holds the Concordia Research Chair Tier I in Information Systems Security. He is the Founder and one of the leaders of the Computer Security Laboratory, Concordia University. He has published three books and more than 260 research papers in journals and conferences on computer security, cyber forensics, privacy, cryptographic protocols, threat intelligence generation, malware analysis, reverse engineering, specification and verification of safety-critical systems, formal methods, Java security and acceleration, programming languages, and type theory. He supervised for the successful completion of 22 Ph.D. students and more than 60 master's students. He is also the President of the National Cyber Forensics Training Alliance, Canada.

RABIA BATOOL received the master's degree in computer science from Kyung Hee University, South Korea. She is currently a Researcher with the College of Technological Innovation, Zayed University. Her research interests include data mining, machine learning, natural language processing, and information extraction, and she has published several research papers in these areas.

ANDREW MARRINGTON received the Ph.D. degree in digital forensics from the Queensland University of Technology (QUT), where he studied at the Information Security Institute.

In 2015, he was appointed as the Acting Associate Dean for Academic Affairs. He was a Research Fellow and a Project Leader with QUT, a Software and Systems Engineer with Queensland Rail, and a Private Sector IT Consultant. He is currently the Associate Dean and an Associate Professor with the College of Technological Innovation, Zayed University. His primary field of research is digital forensics, particularly in automating aspects of the digital forensics process. He is also interested in other aspects of information security. From 2012 to 2015, he served as the College's Graduate Program Director.