A social graph based text mining framework for chat log investigation

Article *in* Digital Investigation · December 2014

DOI: 10.1016/j.diin.2014.10.001

CITATIONS
12

READS
813

2 authors:

Tarique Anwar
Macquarie University
28 PUBLICATIONS **158** CITATIONS
SEE PROFILE

Muhammad Abulaish
Jamia Millia Islamia
88 PUBLICATIONS **594** CITATIONS
SEE PROFILE

A Social Graph Based Text Mining Framework for Chat Log Investigation

Tarique Anwar

Centre for Computing and Engineering Software Systems
Swinburne University of Technology
Melbourne, VIC 3122, Australia
E-mail: tanwar@swin.edu.au

Muhammad Abulaish, *SMIEEE*[*]

Department of Computer Science
Jamia Millia Islamia (A Central University)
Jamia Nagar, New Delhi - 25, India
E-mail: mAbulaish@jmi.ac.in

Abstract

Despite legitimate use, the computer-assisted social media and their features are being exploited by anti-social elements to practice various kinds of cyber-crimes, including hatching plots for criminal activities (like online fraudulence, hacking, drug smuggling, and terrorism) and sneaking into homes to lure or cyber-bully children and opposite sexes. As a result, huge amount of unstructured data are being generated as chat logs. Investigation of such data confiscated from a crime scene goes beyond the scope of manual analysis. This paper presents a unified social graph based text mining framework to identify digital evidences from chat logs data. It considers both users' conversation data and their interactions in group-chats to discover overlapping interests and social ties between them. Instead of applying partial or full parsing, which does not suit informal texts, the proposed framework applies n-gram technique in association with a self-customized hyperlink-induced topic search (HITS) algorithm to identify *key-terms* representing users' interests, *key-users*, and *key-sessions*. We propose a *social graph* generation technique to model user interactions, where the edges (relationships) are established between node (users) pairs only if the corresponding users have participated in at least one common group-chat session, and weighted by the degree of overlap in their interests and interactions. Finally, we present three possible cyber-crime investigation scenarios and devise user-group identification methods for each of them. The user-group identification methods are based on different clustering algorithms having specific properties. We present our experimental results on a dataset comprising 1100 chat logs of 11143 chat sessions continued over a period of 29 months from January 2010 to May 2012. Experimental results suggest that the proposed framework is able to identify key-terms, key-users, key-sessions, and user-groups from chat logs data, all of which are crucial for cyber-crime investigation. The considered chat logs are recovered from a single computer, but in real scenarios it is very likely that logs from more suspects' computers are collected. Combining all the recovered chat logs together would enrich the dataset and thus enrich the relationships in the social graph. Through our experiments, we show that the objective can be achieved even with logs recovered from a single computer by using the group-chat data to draw the relationships between every pair of users.

Keywords: Data mining, digital forensics, social graph generation, chat logs mining, cyber-crime investigation

1. Introduction

The recent advancements in Information and Communication Technologies (ICT) are leading to several new fascinating trends in personal lives. E-communication through chat servers, Instant Messaging (IM) systems, and Internet Relay Chat (IRC) is one of the rapidly growing communication types, which initially remained popular only among the teenagers. However, due to the convenient, instant, and sophisticated nature of broadcasting and exchanging information, these days people from all walks of life find e-communication useful. Instant messaging basically refers to a client-based peer-to-peer chat discussion occurring between a small number of participants wherein the chat traffic is directly transmitted to the clients without any interruption of the server, whereas in server-based chat systems, every chat message passes through dedicated servers that direct it to the respective clients [12]. Millions of users frequently interact with their friends, family members, colleagues, or even strangers to exchange their views, thoughts, and feelings through different IM systems. Windows Live Messenger (previously MSN Instant Messenger), AOL Instant Messenger (AIM), Yahoo Messenger, Google Talk, Skype, and Google+ Hangouts are among the popular freely available IM systems. In response to the widespread user demands, the IM systems are also popular in corporate and government organizations for news updates, notifications, marketing, and many other

[*]To whom correspondence should be addressed. E-mail: mAbulaish@jmi.ac.in, Telefax: +91-11-26980014

activities. This demand has led to the introduction of Enterprise Instant Messaging (EIM) systems that comply with security and legal aspects. IBM Lotus Sametime and Microsoft Lync Server (previously Microsoft office communications server) are the two leading EIM systems.

Most of the IM systems provide a feature to log all instant conversations and maintain history for later reference. To monitor IM chats, some commercial IM monitoring tools like WebWatcher[1] and Spector Pro[2] exist, but they provide limited functionalities. Enrichment of the functionalities of such monitoring tools to facilitate content and interaction analysis at different levels of granularity is an open research problem for the text mining community. In contrast to face-to-face communication, chat communication allows users to anynomize their identity while interacting with others. Unfortunately, this unique feature of social media has proliferated their use among anti-social and criminal persons. Hatching plots for criminal activities (like online fraudulence, hacking, drug smuggling, and terrorism), sneaking into homes to lure or cyber-bully children and opposite sexes by cyber-predators and pedophiles, and committing corporate or homeland espionage are few anti-social activities perpetrated using IM chats in a sophisticated manner by the tech-savvies [12, 13, 5, 33]. In real life, after perpetration of a crime (say a suicide which may be due to cyber-predation or drugs) at some place, the investigation team reaches the spot and investigates every piece of information that could lead to some conclusion. Due to the technological advancements of modern days, several kinds of tech-savvy gadgets are also seized from the spot, e.g., laptops, mobile phones, and memory components (microchips). Investigation of these digital gadgets opens up the research area of cyber-crime investigation. This paper aims to devise a novel technique for mining huge amounts of chat logs recovered from a confiscated computer hard disk and automatically extracting critical crime-related information to assist in the investigation process.

Chat logs are usually stored as HTML files. Each HTML file represents a chat log and contains the record of a set of chat sessions along with the associated meta data. Hence, the complete chat discussion over a period of time is organized as a collection of chat sessions that are participated by the users involved in the conversation. This paper presents a complete framework to mine chat logs by applying a unified text mining approach intended to aid in cyber-crime investigation. It analyzes both user interactions and conversation data together to discover their interaction patterns and overlapping interests. Although the proposed approach is generic for all kinds of chat logs, its experimental evaluation is based on chat logs archived using `Messenger Plus!` and recovered from a confiscated computer hard disk. In summary, the major contributions of this paper are as follows.

- A multi-stage chat logs pre-processing technique, including HTML tag filtering, information component extraction, noise normalization, and slang normalization, to filter out noisy and irrelevant data.
- An n-gram technique to identify candidate key-terms from message contents, and a self-customized HITS algorithm to identify feasible *key-terms*, *key-users*, and *key-sessions*.
- A *social graph* construction technique based on both chat logs meta data and message contents in which nodes represent users and links represent their ties developed through the chat interactions.
- Three possible crime investigation scenarios and a user-group identification method for each of them. *Partitive*, *hierarchical*, and *random-walk* user-group identification methods based on k-means, hierarchical agglomerative clustering, and Markov clustering algorithms, respectively are proposed to identify user groups with overlapping interests and user interactions patterns in different perspectives.

The rest of the paper is organized as follows. Section 2 presents some challenges with mining informal textual communications, followed by some related works in Section 3. Section 4 states the problem addressed in this paper, and the proposed social graph based chat logs mining framework is presented in Section 5. Section 6 presents the experimental results. Finally, Section 7 concludes the paper.

2. Challenges with Informal Communications

In the past, chat logs have been studied for mining digital evidences, but all of them faced the common challenges posed by the noisy and informal nature of textual conversation data. The discourse of these electronic conversations is neither writing nor speech, rather it can be said as written speech or spoken writing or something unique [21]. Their intricate sentence chunks do not follow the grammatical rules and language specific dialects that lead to the failure of language-specific parsers and traditional representation models for information extraction or distillation. Some of the major challenges that inevitably complicate the task of mining textual chat conversation data are summarized below [4, 21, 10, 28]:

Multilinguality: A majority of users in the world are bilingual and they frequently use multiple languages while chatting. In some cases, they use the same Latin alphabet in different languages. For example, English and Dutch both have the same set of letters and it is difficult to differentiate between the words of these languages until a language-specific vocabulary is referenced. Sometimes, different alphabets (e.g., English and Arabic) are also used together in the same chat for better explanation of some specific things. Similarly, while communicating in a language other than English, people quite often use the same Latin alphabet but converse in a different specific language. For example, it's a common habit to converse in Hindi dialect using the Latin alphabet.

Slang and Neologism: During online written conversations, in order to convey or express more using less number of typed

[1] `http://www.webwatchernow.com/Record-Instant-Messages.html`

[2] `http://www.spectorsoft.com/products/SpectorPro_Windows/`

characters, people are habitual to use shortened terms. For example, the word "fine" can be written in just two characters as "f9". There also exist another impulse behind the use of such slang expressions. Since generally people chat online with their fellows, they follow some stylish patterns and transform a standard word into a different one just for the fun and excitement, although it conveys the same meaning. Sometimes, communication patterns keep on continuously evolving and lead to creation of new vocabularies that become the new standard for participating users. For example, "lol" is one such slang, which has now become a standard term and stands for "laugh out loud".

Noise: IM users are always bounded to instantly respond to their partner, and due to this hurrying and casualness they rarely verify the spelling and grammatical correctness of their messages. If a message remains incomprehensible, then it is again elaborated in subsequent messages. Moreover, the stylish behavior of participants make them induce noise into the conversation. For example, in the sentence "I m f999999.....", "m" and "f9" are slang expressions, but the extravagant use of 9s and dots are nothing, but noise.

Intertwined communication threads: During instant messaging, one user responds to what the other user enquired about in the preceding message, and consequently a thread continues to grow. Sometimes, before the first user finishes the response to an earlier message, the other user suddenly starts a different thread of discussion with some new topic in mind. In this case, it becomes unpredictable for an analyst to link messages to a right discussion thread.

Emoticons: Emoticons are visual cues or icons composed of a sequence of letters and punctuation symbols to represent facial expressions. They are generally used in a non-verbal communication to express inside feelings or emotions [35]. The most widely used emoticons ":-)", ";-)", and ":-(" represent a smile, a wink, and a frown expression, respectively. Determining users' sentiments and moods from emoticons is a new add-on to the user-generated-contents-related text mining research problems.

Most of the IM-related practices mentioned above are also very common in other forms of social media communications like blogs, microblogs, forums, and social networking sites, which urges the attention of text mining researchers to devise generic and scalable techniques for handling and analyzing the noisy unstructured textual data.

3. Related Work

Communication through computer-assisted social media have remained under a substantial study over the last few years. Due to increasing popularity, both structural and textual content of social networks generated through emails, blogs, microblogs, chats, forums, opinion sources, Social Networking Sites (SNSs), and other broadcasting mediums are being studied in different perspectives. In the context of business intelligence, corporate and profit-making organizations are considering User-Generated Contents (UGC) for user-centric research to identify users' interests for their products in the market [19]. Marketing products through targeted influential users in social networks to generate a word-of-mouth is an important area in marketing research. News mining from real-time users' comments is emerging as a significant application of UGC [22]. Studying a nation's political stabilization [25] and disaster management during natural hazards [30] are other few areas where social media applications have found their place.

In addition to the use of computer-mediated communication media for social interactions, the last decade has detected traces and several cases of their illegitimate use for crime purposes [8, 13, 5]. Several prior works have highlighted this aspect and proposed potential solutions to track anti-social activities over the Web. User interaction patterns play a significant role in these works, which is followed by the context of their interaction. In [14], Chu *et al.* explored the role of SNSs in digital crime investigation. For a crime related to `Facebook`, they proposed a methodology to collect and analyze digital evidences via live internal data acquisition, and systematically reconstruct the crime scene with respect to previous `Facebook` sessions of the victim. In [8], we studied the practice of racism and extremism in Web forums and proposed an algorithm to identify cliques. Beebe and Clark [11] proposed a text string search methodology intended to improve the information retrieval effectiveness of digital forensic text string searches and aid digital investigators in quickly locating hits relevant to the investigative objectives. They grouped the digital forensic text string search results by applying a post-retrieval clustering algorithm, specifically by using Kohonen self-organizing maps. Louis and Engelbrecht [23] assessed the usefulness of text mining approaches in evidence discovery for crime investigation and proposed an unsupervised framework for relation extraction from a text corpus.

Chat logs have remained under study in various perspectives [12] [28] [16] [21] [26] [13] [5] [17]. Because of its importance in multiple research and applied communities, Uthus and Aha recently did a survey on chat analysis research [33]. They categorized the research in this area into low-level and high-level analysis, where the low-level problems of study mainly include chat-preprocessing [12] [28], thread disentanglement [15], and chat room feature preprocessing [3] [15], and the high-level problems mainly include topic detection [3] [26], message attribute identification [29] [21], social phenomenon detection [32] [13], automatic summarization [31], and user profiling [12]. Intended to assist in crime detection, `ChatTrack` [12] automatically profiles the topics of discussion in a particular chat room or by a particular individual. It follows a classification technique that needs to train the system on a known data set. Security personnel can focus their attention on people or discussions violating the ethical codes of conduct. Schmidt and Stone [28] studied the noisy nature of texts in IRC chat logs and explored existing techniques for topic-wise segmentation of text messages. To detect a topic change, they proposed a system for text segmentation, which is based on text tiling, pause detection, and latent sematic analysis. Holmer [16] went through a discourse analysis to deal with the intertwined structure of discussion threads. In [21], Kucukyilmaz *et al.* examined the predictability of user- and message-specific attributes from real-time online chat messages. They used a term-based

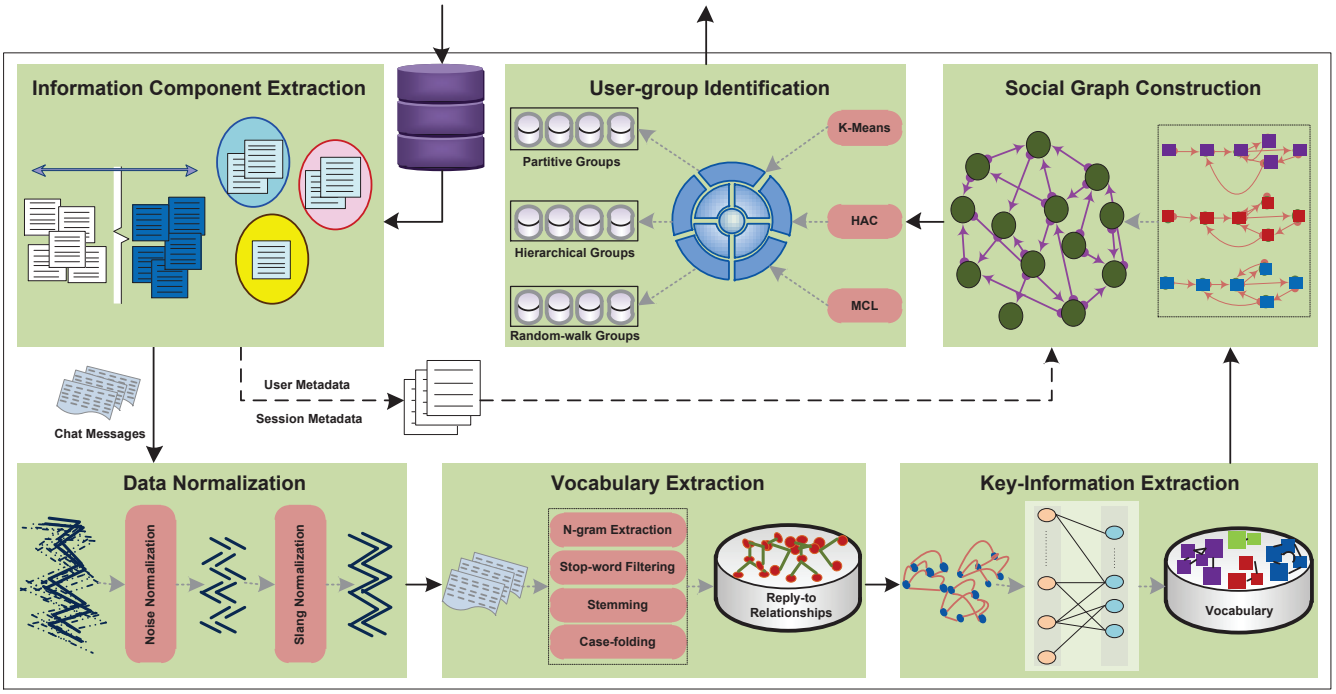


Figure 1: Proposed social graph based text mining framework for chat log investigation

approach to investigate user and message attributes in the context of vocabulary use and a style-based approach to examine authors' writing styles. Both of them accept it as a classification problem and apply machine learning with various known classifiers. Özyurt and Köse [26] designed a supervised learning approach for identification of discussion topic from chat messages, specifically in Turkish language. Bogdanova *et al.* [13] addressed the problem of automatic detection of sexual predators and pedophiles by mining sentiments and a set of content-based features. The recent works [5] and [17] mined criminal networks to aid in digital crime investigation and forensic analysis of unstructured textual data. Al-Zaidy *et al.* [5] proposed a semantic data mining method to discover criminal communities, identify their prominence, capture the concealed relationships among them, and extract information that could potentially lead to track the criminals. Iqbal *et al.* [17] worked on the problems of user clique mining from chat logs and topic analysis to identify the cliques with criminal activities involvement. Both these methods are heavily dependent on language-specific linguistic features, as they use named-entity recognizer (NER), open text summarizer, and other assumptions specific to English language. Following this approach sometimes contradicts for handling unstructured texts of user conversations, specifically chat messages, unless there is rigorous pre-processing to transform them into texts complying with English grammatical rules and usage. However, text normalization in itself is a major issue for noisy texts and it is still at infancy. In the proposed work, we overcome this problem by considering weighted *key-terms* instead of entities extracted by an NER.

4. Problem Statement

Let us suppose a forensic investigator seizes a computer (or a set of computers) from a crime scene or a suspect. Usually there exist various kinds of data on the computer hard disk. Let a set of chat logs from an instant messaging system is recovered, which presumably contains the past discussions among the suspects. The investigator wants to find more information about the crime and the crime suspects through the chat logs. As the textual conversations in the chat logs remain huge in size, it is beyond the scope of manual investigation to go through the vast logs and inter-relate everything properly. The problem addressed in this paper intends to mine such chat logs recovered from one or more computers for cyber-crime investigation. We formally present it as designing a framework for investigation of suspicious chat logs, which primarily consists of two sub-problems mentioned below.

4.1. Key-Information Extraction

This problem deals with the extraction of the key information components that provide details about the suspects and the crime. We identify sets of ranked *key-terms*, ranked *key-users*, and ranked *key-sessions*. The key-terms are the highly informative terms conversed among the suspects which give clue about the crime. The key-users are the leading users having a dominating role in the conversation, and a similar role can be inferred in the crime. The key-sessions are the most important chat sessions in which crucial discussions took place.

4.2. Identification of Different User-Groups

For a proper investigation, it is very important to identify how the chat users are related among themselves. It is also crucial to

identify different closely associated groups among the chat participants. After having the list of ranked key-users, this problem deals with identifying the grouping patterns among them based on their overlapping interests and interactions. We propose three different methods to identify user-groups with frequent interactions and overlapping interests, where each method handles a specific crime investigation scenario, described in Section 5.5.

5. Proposed Social Graph Based Text Mining Framework

The proposed framework is designed to be applicable in a situation where chat logs from a suspect's (among potentially a group of suspects) computer are recovered by the crime-investigation personnel. These chat logs would contain only those sessions in which the known suspect participated, according to which this suspect plays the central role in interactions with all the other users. However, many of those chats are done in groups, where the sessions involve more than two participants. In fact, group chatting is quite a common activity in instant messaging. In our experimental dataset, most of the chat sessions involved at least three participants. We use these group chats to draw the relationship between hundreds of the other users. Their relationship strength is also based on the textual contents of their comments. The interaction pattern in this scenario does not form a clear star topology; rather the edges (interactions) are distributed in between all the nodes (participating users) in the graph. Our experiments are based on chat logs recovered from a single computer, but in real scenarios it is very likely that logs from more suspects' computers are collected. Combining all the recovered chat logs together would enrich the dataset and thus enrich the social graph (especially the relationships between different users) along with other results. Through our experiments, we show that the objective can be achieved even with logs recovered from a single computer, may be with some compromising results.

The framework unifies user interaction and conversation data together to identify key information components and different user-groups. Figure 1 presents the architecture of the proposed framework for chat log investigation, which performs five different tasks– *i*) data extraction and normalization, *ii*) vocabulary extraction, *iii*) key-information extraction, *iv*) social graph construction, and *v*) user-group identification. First of all, the chat logs are processed to identify different information components and normalize them for noise removal and slang neutralization. The second step applies an n-gram technique to extract a *vocabulary* set of the chat community, which is followed by the extraction of key-information and computation of feature values by constructing two bipartite graphs and applying HITS on them. Thereafter a social graph of users is constructed as a weighted graph using their interaction patterns in the group chat sessions. Finally, the social graph is used to identify three different kinds of user-groups using *k*-means, hierarchical agglomerative clustering, and Markov clustering techniques. We present three different crime investigation scenarios, and each user-group identification method is specifically designed to address one of the scenario. Further details about these tasks are presented in the following sub-sections.

```
<body>
    <h1>Messenger Plus! Chat Log</h1>
    <div class="mplsession" id="Session_2010-04-04T13-37-44">
    <h2>Session Start: Sunday, April 04, 2010</h2>
    <ul>
        <li class="in">McKrautney. <span>(jesse.hutton@hotmail.com)</span></li>
        <li>Cindy <span>(aerosmith_tyler@msn.com)</span></li>
    </ul>
    <table cellspacing="0">
    <tbody>
        <tr><th><span class="time">(1:37 PM)</span> Cindy:</th>
              <td style="font-family:&amp;quot;Dotum&amp;quot;;font-weight:bold;">Heey.</td>
        </tr>
        ⋮
    </tbody>
    </table>
    </div>
        ⋮
    <div class="mplsession" id="Session_2010-04-04T13-37-44">
        ⋮
    </div>
</body>
```

Figure 2: A sample format of the chat logs archived through `Messenger Plus!`

5.1. Data Extraction and Normalization

This task aims to transform raw chat logs into a machine-readable format. It consists of three sub-tasks that are described in the following sub-sections.

5.1.1. Information Component Extraction

Chat messages are logged in various formats depending on the application platforms and their settings. The data under study is a collection of chat logs archived through `Messenger Plus!`, which is a third party extension of Windows live messenger. The logs are available as HTML files, each of which contains discussions of one or more chat sessions. Figure 2 shows a snippet of a typical log file, in which each session is marked with ⟨`div`⟩ tag and meta data is stored in id attribute. participants of a chat session are marked with ⟨`ul`⟩ tag, inside which ⟨`li`⟩ tag marks users' name and email id. Some other tags highlight information components as shown in Figure 2. Considering these tags as markers, all information components are extracted from the HTML files, which includes session start date and time, username and email id of participating users, chat messages and their posting time. Out of all these information components, only the chat messages lack a standard form. The issues with noisy texts discussed in Section 1 make the chat messages unfit for their use in raw form. Hence, the proposed method goes through the steps of noise and slang normalization to normalize the chat messages into standard format.

5.1.2. Noise Normalization

The most common form of noise in chat messages is the unnecessary repeated use of punctuation marks or letters. The repeat may occur at any position – start, middle, or end of a word. For example, "okkkk" and "really?????" have single character repeat at the end, "hahaha" has double-character repeats, and "wowwowwow" has triple-character repeats. We use regular expressions to normalize all such kind of repeats. To normalize

the words containing punctuation marks and digit-letter mixing, we assume that no character can repeat more than once continuously, and consequently the excessive characters are dropped, such that "f99" becomes "f9" which is a slang version of "fine". For letters, we assume that they can not repeat continuously more than twice, and therefore drop the extra repeats, such that "oookkkk" becomes "ok", "okk" remains as it is, "freakkkky" becomes "freaky", "freaeakkkky" becomes "freaky", and "add" also remains as it is.

5.1.3. Slang Normalization

Slang expressions commonly used in chat messages have no booked place in standard dictionaries. Therefore, we compiled a list of slang expressions and their equivalent standard terms from different sources and personal surveys. We follow a table-lookup process to scan the complete set of chat messages to identify slang expressions and replace them with the equivalent standard terms. For example, "⟨f9, fine⟩" replaces each occurrence of "f9" by "fine", and "⟨lol, Laugh out loud⟩" replaces each occurrence of "lol" by "Laugh out loud".

5.2. Vocabulary Extraction

In general, a *vocabulary* is a set of terms or lexicons that completely cover a user's or a community's communicative knowledge. In context of chat communications, we define *vocabulary* as a set of valuable information containing key-terms exchanged among participating users during its complete *life of communication*[3]. Therefore, the vocabulary extraction process aims to identify vocabulary of the community involved in chat discussions, and it comprises four sub-tasks – *n-gram extraction*, *stop-word removal*, *stemming*, and *case-folding*. An *n-gram* is defined as a sequence of n consecutive words in a chunk of text [2]. Depending on the value of n, it can be a 1-gram containing single word if n is 1; 2-gram containing two consecutive words if n is 2; and so on. Since a key-term usually comprises three words atmost and exceeding this is an exception, the proposed method generates chunks of texts from chat messages by a process called *tokenization* (described in our earlier works [1, 2]), and then extracts all possible 1-, 2-, and 3-grams from them. Those beginning or ending with a stop-word are usually not complete in their information component. Therefore, all such n-grams are filtered out from the list. Hence, an 1-gram is discarded if it is a stop-word, a 2-gram is discarded if any of its constituting words is a stop-word, and a 3-gram is discarded if any of the boundary words is a stop-word. Moreover, a single word can have several different forms (e.g., "laugh" and "laughing"), which although convey the same meaning but do not match exactly. To neutralize this affect, all n-grams are stemmed using Porter Stemmer [27] to consider only their base forms. Similarly, a word can be differently cased, which again leads to a mismatch for same words. To neutralize this affect all n-grams are case-folded.

[3] By *life of communication*, we mean all chat sessions extracted from a single confiscated disk.

5.3. Key-Information Extraction

This task aims to extract key information components from chat logs and to compute their feature values, where the key information refers to three things – vocabulary terms, participating users, and chat sessions. The computed feature values define their leading roles, which can also be used to rank them. The usage of vocabulary terms by chat participants follow different patterns. Each one has some specific level of prominence or implication in the whole chat discussion. By using statistical and computational techniques, this step extracts feature values for all terms existing in the extracted vocabulary to characterize their prominence in the whole chat discussion. There exist some terms that are frequently used, while others not. The vocabulary usage pattern remains specific to two dimensions – *participating users* and *chat session*. Every user roughly follows a pattern of vocabulary usage unintentionally; and since a chat session includes discussion at a specific point of time and situation, it remains confined to a specific vocabulary centered around the topic of discussion. Thus, there exist two kinds of relations in usage patterns – *vocabulary–user* relation and *vocabulary–session* relation. To explore these relationships further, a bipartite graph is constructed, which is treated by a self-customized Hyperlink-Induced Topic Search (HITS) algorithm [20] to compute hub and authority scores. HITS algorithm distinguishes hubs and authorities in the set of objects. A hub object has links to many good authorities and an authority object has a high quality content with many hubs linking to it. The hub and authority scores are computed in an iterative manner.

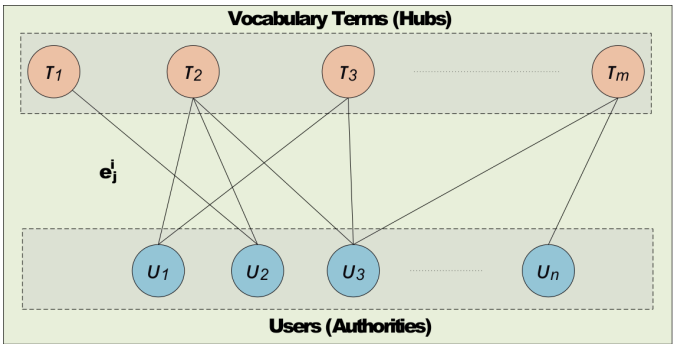


Figure 3: Term–user bipartite graph

To start with, a *term–user* bipartite graph is constructed, considering terms in the vocabulary as hubs and users as authorities. A user node or authority is linked to all those term nodes or hubs that have been used by the user at least once in a chat message. On the other hand, a term node or hub is linked to all those users or authorities who have used it at least once in a chat message. Formally, the bipartite graph is represented as a triplet of the form $G^{TU} = (V_\tau, V_\nu, E_{\tau\nu})$, where $V_\tau = \{\tau_i\}$ is the set of terms in the vocabulary, $V_\nu = \{\nu_j\}$ is the set of participating users, and $E_{\tau\nu} = \{e_i^j | \tau_i \in V_\tau, \nu_j \in V_\nu\}$ refers to the correlation between vocabulary terms and users. Each edge e_i^j is assigned a weight $\omega_i^j \in [0, 1]$ to represent the strength or integrity of a

relationship between a term τ_i and a user ν_j. The weight ω_i^j of a term τ_i associated with a user ν_j in chat sessions is calculated using equations 1 [6], where $\text{freq}(\tau_i, \nu_j)$ denotes the number of times term τ_i is used by user ν_j, $|\nu_j|$ denotes the total number of terms in the vocabulary that has been used at least once by the user ν_j, $\overline{\nu}$ denotes the average number of terms associated with a user, $|\nu|$ denotes the total number of participating users in the complete set of chat sessions, and $\text{ifreq}(\tau_i, \nu)$ denotes the inverse user frequency of τ_i in the set ν.

$$\omega_i^j = \left(\frac{\text{freq}(\tau_i, \nu_j)}{\text{freq}(\tau_i, \nu_j) + 0.5 + (1.5 \times \frac{|\nu_j|}{\overline{\nu}})} \right) \times \left(\frac{\log \frac{|\nu|+0.5}{\text{ifreq}(\tau_i, \nu)}}{\log(|\nu| + 1)} \right) \quad (1)$$

Authority score $AS^{(t+1)}(\nu_j)$ for user ν_j and hub score $HS^{(t+1)}(\tau_i)$ for term τ_i in $(t+1)^{th}$ iteration are based on the authority and hub scores obtained during t^{th} iteration and calculated using equations 2 and 3, respectively. After each iteration, authority and hub scores are normalized by dividing them by the corresponding *norm* measures defined in equations 4 and 5, respectively.

$$AS^{(t+1)}(\nu_j) = \sum_{\tau_i \in V_\tau} \omega_i^j \times HS^{(t)}(\tau_i) \quad (2)$$

$$HS^{(t+1)}(\tau_i) = \sum_{\nu_j \in V_\nu} \omega_i^j \times AS^{(t)}(\nu_j) \quad (3)$$

$$norm_{AS} = \sqrt{\sum_i \left(AS^{(t)}(\nu_i) \right)^2} \quad (4)$$

$$norm_{HS} = \sqrt{\sum_i \left(HS^{(t)}(\tau_i) \right)^2} \quad (5)$$

The bipartite graph G^{TU} represented using adjacency matrix is given below, where, $\vec{a}^{(t)} = [AS^{(t)}(\nu_j)]_{|V_\nu| \times 1}$ is the vector of authority scores for users in the t^{th} iteration, and $\vec{h}^{(t)} = [HS^{(t)}(\tau_i)]_{|V_\tau| \times 1}$ is the vector of hub scores for the terms in t^{th} iteration.

$$L = (L_{i,j})_{|V_\tau| \times |V_\nu|} \quad (6)$$

such that:

$$\vec{a}^{(t+1)} = L\vec{h}^{(t)} \quad (7)$$

and

$$\vec{h}^{(t+1)} = L\vec{a}^{(t)} \quad (8)$$

The iteration process continues until convergence is achieved, i.e., until the difference between two successive *norm* measures falls below 0.0001. After convergence, the resultant hub scores of vocabulary terms τ_i and the authority scores of users ν_j are considered as their feature values, μ_{τ_i} and μ_{ν_j}, respectively. Based on μ_{τ_i} values, the terms in V_τ are sorted and top-ranked terms are declared as *key-terms* representing the main theme of the whole chat discussion. Similarly, based on μ_{ν_i} values, users in V_ν are sorted and top-ranked users are declared as *key-users* playing leading roles in the discussion.

In the second phase of feature extraction process, another bipartite graph G^{TS} is constructed, considering vocabulary terms as hubs and chat sessions as authorities. Formally, it is represented as a triplet of the form $G^{TS} = \left(V_\tau, V_\xi, E_{\tau\xi}\right)$, where $V_\tau = \{\tau_i\}$ is the set of vocabulary terms, $V_\nu = \{\xi_j\}$ is the set of chat sessions, and $E_{\tau\xi} = \{e_i^j | \tau_i \in V_\tau, \xi_j \in V_\xi\}$ refers to the correlation between users and vocabulary terms. The weight $\omega_i^j \in [0, 1]$ of an edge e_i^j is calculated in the same way as Equation 1, except that the measures are computed with respect to the sessions ξ instead of users ν. HITS algorithm is applied on G^{TS} in the same way as earlier, and final authority and hub scores are considered as feature values, μ_{τ_i} and μ_{ξ_j}, for terms and sessions, respectively. On sorting the vocabulary terms based on μ_{τ_i} values, we get another set of *key-terms* with respect to sessions. Based on μ_{ν_i} values, chat sessions are sorted and the top-ranked sessions are declared as *key-sessions*. They are considered as the most important discussions with respect to their coverage through vocabulary terms.

Based on the two different sets of feature scores for vocabulary terms, final score μ_{τ_i} for each term τ_i is computed using Equation 9, where $\mu_{\tau_i}^{TU}$ and $\mu_{\tau_i}^{TS}$ are the feature scores computed during the first and second phase, respectively, of the feature extraction process, and $\alpha \in [0, 1]$ is a constant.

$$\mu_{\tau_i} = \alpha \times \mu_{\tau_i}^{TU} + (1 - \alpha) \times \mu_{\tau_i}^{TS} \quad (9)$$

5.4. Social Graph Construction

A chat session generally contains a group of users interacting with each other, and such interactions establish a kind of *tie* or *bond* between them. The motive behind social graph construction is to model the participating users and their interaction patterns into a rich structure which could represent the ties among the participating users. We model social graph as a weighted graph $G = (V_\nu, E_{\nu\nu}, W_{\nu\nu})$, where $V_\nu = \{\nu_i\}$ is the set of nodes representing all participating users, $E_{\nu\nu} \subseteq V_\nu \times V_\nu$ is the set of edges representing ties among the users, and $W_{\nu\nu} = [0, 1]$ is the set of weights assigned to edges. An edge between a pair of users ν_i and ν_j, e_i^j is created if they participate together in at least one chat session. Considering top-k key-terms based on their feature scores, each user $\nu_i \in V_\nu$ is assigned a feature vector $\vec{\Phi}_{\nu_i} = \langle \phi_1^{\nu_i}, \phi_2^{\nu_i}, \ldots, \phi_k^{\nu_i} \rangle$, where $\phi_j^{\nu_i} = \mu_{\tau_j}$ if a term τ_j has been used by user ν_i at least once, otherwise the value of $\phi_j^{\nu_i}$ is set to 0. Thereafter, weight of an edge connecting a pair of users ν_i and ν_j, w_i^j, is calculated using Equation 10, where $\deg(\nu_i, \nu_j)$ is the number of sessions in which both of the users participated together, and $\deg(\nu_i)$ and $\deg(\nu_j)$ are the degrees of the nodes corresponding to the users ν_i and ν_j, respectively in the social graph.

$$w_i^j = \frac{\Phi_{\nu_i} \cdot \Phi_{\nu_j}}{\left|\Phi_{\nu_i}\right| \left|\Phi_{\nu_j}\right|} \times \frac{\deg(\nu_i, \nu_j) \times \left(\deg(\nu_i) + \deg(\nu_j)\right)}{2 \times \deg(\nu_i) \times \deg(\nu_j)} \quad (10)$$

The weight calculation formula beautifully captures two different different types of data (overlapping interests and interactions) at the same time. It considers textual conversation

data representing users' interests in the first part and interaction structure data in the second part of the formula, and multiply them together to get the final weight. The first part computes the cosine similarity between two feature vectors, where a feature vector consisting of feature values of the key-terms conversed by the respective users. Its value range from 0 (if the vectors are completely dissimilar) to 1 (if the vectors are exactly same). The second part of the formula determines the tie (or the degree of association) between a pair of users by considering their interaction pattern in the chat sessions. Its value range from 0 (if a pair of users never shared any chat session) to 1 (if a pair of users always participated together in the chat sessions). Ultimately, the final weight ranges from 0 to 1.

5.5. User-Group Identification

The social graph constructed in the previous subsection contains a precise information of the tie between the users. In this subsection, we use that information to identify different user groups in different crime scenarios. We identified three different scenarios such that any crime investigation scene can fit in one of them.

- **Scenario 1**: The investigators already have the information about the exact number of user groups, which may have been derived from the suspects' affiliations, organizations, regions, or other related networks.
- **Scenario 2**: The investigators have partial information about user groups due to which the users can't be grouped directly, rather they can be organized in the form of a hierarchy. For example, a user (suspected criminal) may be affiliated to some organization which follows a hierarchy for organizing crimes.
- **Scenario 3**: The investigators have absolutely no clue about the user grouping pattern.

Clustering is a well known technique for grouping objects. It is defined as an unsupervised learning process to organize unlabeled objects into different groups or clusters, members of which are similar in some way [18, 37]. Thus, a *cluster* can be considered as a collection of objects that are similar among themselves, but dissimilar to the rest of the objects. Cluster analysis has a significant contribution in data mining, and various techniques have been studied in the past. In this research, we apply clustering on the generated social graph to identify various user-groups. As we have three different scenarios, specific class of clustering algorithms need to be applied to handle them specifically. For each scenario, we identify a specific set of user-groups listed below.

1. **Partitive user-groups**: Partitive methods are a very popular class of clustering algorithms in which a set of objects are clustered into a pre-determined and fixed number of groups based on the pair-wise distance matrix [36]. K-means is the simplest and most popular partitive algorithm belonging to this class. It aims to group data objects into k clusters based on the similarity of their feature values, where the value of k needs to be supplied as an input. Considering a set of n data objects $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$, where each d_i has m feature values forming an m-dimensional vector $\langle f_1, f_2, \ldots, f_m \rangle$, the k-means algorithm aims to find the subsets $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k\}$ that minimizes $\sum_{l=1}^{k} \sum_{d_i \in \mathcal{D}_l} dist(d_i, \mu_l)$, where μ_l is the mean value of subset $\mathcal{D}_l$, and $dist()$ is the function that computes the distance between two vectors. The method starts with randomly picking k data objects for initial k number of means. Thereafter, remaining data objects are compared with the means of each group and assigned to the closest one, and the means are updated. The mean update and object assignment processes continue iteratively until convergence. To deal with the investigation scenario 1, we apply k-means on the set of users. i.e., each $d_i \in \mathcal{D}$ is a participating user, and the distance matrix is derived from the adjacency matrix A_G of the social graph G. After convergence of the k-means algorithm, the set of user-groups obtained is termed as *partitive user-groups*.
2. **Hierarchical user-groups**: Hierarchical clustering algorithms are another class of clustering algorithms in which a set of objects are organized into a hierarchical structure according to the affinity matrix [37]. It groups the objects into different sets of clusters at different levels of granularity based on the affinity strength between them. The hierarchical agglomerative clustering (HAC) is a popular and efficient algorithm belonging to this class. For a set of n data objects, it first computes an affinity or similarity matrix $A_{n \times n}$, and starts with keeping each data object in a separate cluster. Then it goes through $n - 1$ steps of iteratively merging the currently closest pair of clusters to form a single cluster. After each merge, the rows and columns of the merged cluster are updated, and the clustering is stored as a list of merges in A.
   To deal with investigation scenario 2, we apply HAC on the set of users, where the affinity matrix refers directly to the adjacency matrix of the social graph. The set of user-groups thus obtained is termed as *hierarchical user-groups*. It generates user groups at different hierarchies defined by the levels of granularity.
3. **Random-walk user-groups**: Graph clustering algorithms are another class of clustering algorithms that identify the grouping patterns by simulating a random walk in the graph, where the move of walk is influenced by the affinity strength between the nodes. Markov clustering (MCL) algorithm belongs to this class. It is a graph clustering method based on simulating a random walk on a weighted graph [34] [37]. It considers the transition from one node to another within a cluster as much more likely than those in different clusters, taking into account the weight of those links. It accepts the adjacency matrix A_G of a graph G as an input and starts working by adding loops or self-edges to A by setting the diagonal elements as 1, $a_{pp} \leftarrow 1$, if they do not exist, and then converting this matrix into a *Markov matrix* $M_{n \times n}$. M acts as a transition matrix for a Markov chain or a Markov random walk on G. The rest

of the algorithm is an iterative method interleaving matrix expansion by multiplying with itself, $M_i = M_{i-1} \times M_{i-1}$, and applying the inflation operator on M_i using Equation 11. It keeps on iterating until the transition matrix M_i converges. After its convergence in the i^{th} iteration, it results to a directed graph with weakly connected components. The nodes having values greater than zero in the diagonal, i.e., $m_{pp_i} > 0$, are called as *attractors* of the corresponding cluster. All other nodes having a link with the attractor are attracted towards it and are included in that cluster. If a node is attracted towards multiple attractors then there is an overlapping of those clusters to which these multiple attractors correspond. Most of the clustering algorithms need to provide the required number of clusters k as an input parameter, which is often difficult to determine beforehand. Unlike them, MCL is free from this limitation, instead it uses an inflation parameter $r > 1$ to decide the value of k.

$$\xi(M_i, r) = \left\{ \frac{\left(m_{pq}\right)^r}{\sum_{a=1}^{n} \left(m_{pa}\right)^r} \right\}_{p,q=1}^{n} \tag{11}$$

To deal with investigation scenario 3, we apply MCL on the social graph directly, and the set of user-groups obtained is termed as *random-walk user-groups*.

Thus all three crime scenarios are addressed specifically by the different kinds of user-groups described above.

6. Experiments

6.1. Test Bed

This section presents our experimental study on a data set[4] comprising MSN chat logs archived using `Messenger Plus!`. It contains 1100 chat logs of 11143 chat sessions continued over a period of 29 months from January 2010 to May 2012. The total size of the dataset is around 715 MB. It has 733183 commented messages; on an average, 66 messages per chat session. A chat log contains one or more chat sessions and named as one of the participants in the logged chat sessions. Log files are organized in folders named by the month and year (e.g., January 2010) in which the sessions took place.

6.2. Experimental Results

Experiment starts with data extraction from the log files, which goes through HTML tag filtering, and noise and slang normalization. Each log is composed of multiple chat sessions, and each session starts with listing the participating users' names and email ids who initiate the session. These listings are used to generate meta-data about users. In many cases, same email id is listed with multiple usernames, which infer that all those usernames are alias names of a single user. In the context of a cyber-crime investigation, alias names have a significant role in tracking suspects [7, 9]. Table 1 shows a list of alias names of a user having email id `je***.****on@hotmail.com` who has commented 59275 times in the entire discussion using one of these alias names. To maintain the user privacy, we have anonymized the usernames and email addresses. There are also some cases in which different email ids have been used with same username in different chat sessions. This indicates a possibility of the existence of a single person using different email ids. In addition, some times users join a chat session in middle. In this situation, a username has no record of its corresponding email id. Therefore, for usernames associated with an email id, unique users are identified based on e-mail ids, and for rest of the cases unique users are identified using the usernames. In this way, we identified a total of 456 unique users in the entire discussion. During a session, 993 times usernames were changed, 39577 video calls took place, 8574 times users were added to a session after it has already been started, 8069 times users left a session in the middle, and 1759 times personal messages were changed.

Table 1: Anonymized alias names of the user with e-mail id je***.****on@hotmail.com

Mc******ey.	Mc*******er
Et******ft	(L)M*******ey.
Je*******tt!	Sh**********de!
Je*******tt(Mc****zy)	Il**********ck
FA*********RD	=W=(!)Mc****************tt!
Mc****zy	Je*******tt
je***.****on@hotmail.com	

Table 2: Top-10 key-terms

Key-term	Hub score	Key-term	Hub score
wiping cloth	0.7415	butterball	0.6440
summer buddies	0.6325	mayonaise	0.6325
yeait	0.6066	kind ofwant	0.5823
cloth	0.5725	sydney	0.5724
miki stopped	0.5715	secret cat hole	0.5637

Noises in text messages are normalized using regular expressions, and slang expressions are normalized using a compiled list of 620 ⟨slang, term⟩ pairs. The list contains two different categories of pairs. First category contains those terms in which every single character stands for a word or it's simply a slang acronym, e.g., ⟨2U2, to you too⟩. Second category contains other kind of terms in which the group of characters in the whole word stands for a standard word, e.g., ⟨abt, about⟩ and ⟨f9, fine⟩. Emoticons are much about emotions of users during interactions and they play an important role in sentiment analysis. Since the motive behind processing the content of chat messages is to identify feasible key-terms representing users' interests, we filter out the emoticons from further considerations.

Processed chat messages are then subjected to vocabulary extraction, which resulted into the extraction of 1105097 unique vocabulary terms. For feature extraction, two bipartite graphs –

[4]The dataset is publicly available at `http://www.4shared.com/folder/zNTBEp52/_online.html`

Table 3: Top-10 key-users (anonymized)

Email id	Username	#Comments	Authority score
fa*******sh4@hotmail.com	el****aa	8635	0.7615818324
mi*********-***os@hotmail.com	M**i	4401	0.6478153469
an****_z@hotmail.com	a**a	5987	0.0155792762
xxp*******nxx@hotmail.com	-'Jo****;*******al:	1793	0.0092510516
dr.***up@hotmail.com	co**********ng.	1451	0.0003047137
⟨*unavailable*⟩	St**********pi	1295	0.0000116569
⟨*unavailable*⟩	Jo**********ld	2018	0.0000092259
si***_**gs@hotmail.com	Me***.*****.*****(tu)	2402	0.0000049257
ci*****n.22@hotmail.com	ju**a[:	2311	0.0000033987
ja***_**m18@hotmail.com	J***n	5594	0.0000014583

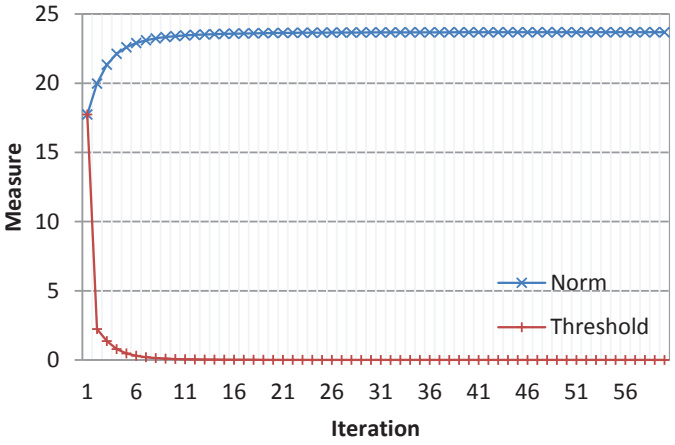


Figure 4: *Norm* and *threshold* values during convergence of term-user bipartite graph

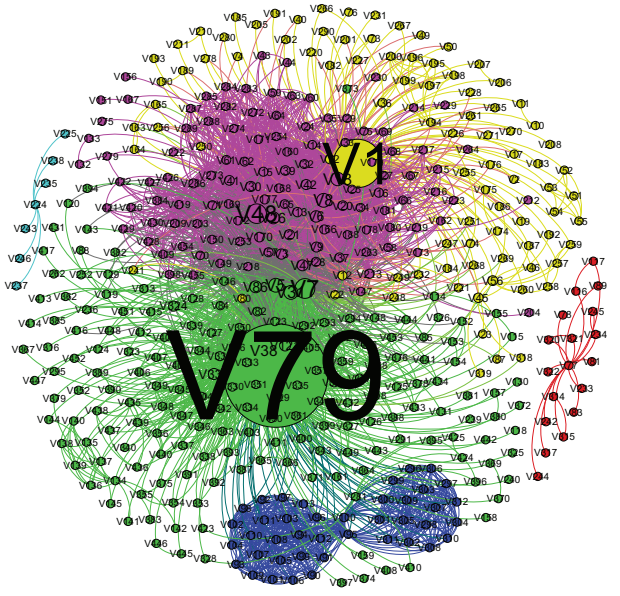


Figure 5: A snapshot of the generated social graph

one for terms (1105097 nodes) and users (456 nodes), and second for terms (1105097 nodes) and sessions (11143 nodes) are constructed. Customized HITS algorithm is applied on both bipartite graphs, which converged in 61 and 67 iterations, respectively. The threshold value for convergence, i.e., the difference between norm of two successive iterations, is set to 0.0001. As a result, we obtained two sets of key-terms, one set of key-users, and one set of key-sessions. Table 2 shows top-10 key-terms along with their final hub scores, and table 3 shows top-10 key-users along with one of their usernames, total number of comments, and final authority scores. On analysis, we found that just by commenting more times does not necessarily make a user more important, dominating, and leading in a user group. There are two users with ⟨*unavailable*⟩ email ids. These users do not have their email id logged in the archived logs, because they were not present at the initiation of sessions, rather entered in the middle. The feature scores of the vocabulary terms are finally computed as discussed in Section 5.3.

Thereafter the social graph is constructed. Its visualization layout is generated using the F Reingold algorithm and shown in Figure 5. In the social graph, nodes correspond to the users and edges correspond to the ties (developed through the entire chat discussion) between the connected users. It can be observed in Figure 5 that $V79$ is the most actively interacting user (supposedly the user from whose computer the chat logs are collected) followed by $V1$, $V48$, and $V7$. Node $V79$ corresponds to `mc******ey@hotmail.ca`, $V1$ to `je****.****on@hotmail.com`, $V48$ to `je******sz@hotmail.com`, and $V7$ to `jo*****im@hotmail.com`. The social graph also contains some disconnected groups. For example, the subgraph in red color in the right. As the chat logs are collected from a single computer, the owner of that computer must have participated in all the logged chat sessions. If it is so, then the the graph should not be disconnected. From a crime investigation point of view, these two contradictory things give an important indication that the user, from whose computer the chat logs are collected, used multiple user accounts to communicate with different user groups. The social graph is finally subjected to clustering algorithms to group users into different clusters and identify their interaction patterns. We identify partitive user-groups by applying k-means algorithm, hierarchical user-groups by applying HAC algorithm, and random-walk user-groups by applying

Markov clustering (MCL) algorithm on the social graph.

6.2.1. Partitive User-Groups Identification

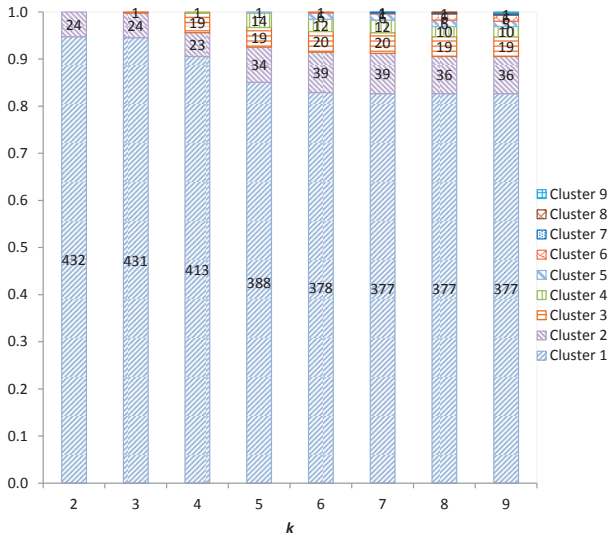


Figure 6: Partitive clusters

The simplest and primary kind of user interaction analysis is to group users into a fixed number of partitions based on their commonalities and analyze their interactions in each partition. k-means is an unsupervised, non-deterministic, and iterative clustering algorithm that generates a pre-defined number of disjoint and non-hierarchical clusters following a partitive approach [24]. We apply a robust version of k-means, partitioning around medoids (PAM), on the user social graph, and the generated clusters are pictorially shown in Figure 6 as a bar chart. For each value of k from 2 to 9 there is a bar and each kind of shades represent a cluster as defined by the legends. The figure highlights the portion of the total number of users lying in each cluster as k increases. A total number of 456 users are grouped into clusters. At $k = 2$, it groups 432 users into the first cluster and remaining 24 users into the other; at $k = 3$ the groupings are for 431 users, 24 users, and 1 user; and at $k = 9$ the clusters are of 377 users, 36 users, 19 users, 10 users, 6 users, 5 users, and 1 user (three clusters). Figure 7 presents the results in terms of their Silhouette measures at each discrete value of k ranging from 2 to 9. Investigation through these user-groups has the potential to discover facts in scenarios when the investigators have information about the number of user groups that may have been derived from their suspected affiliations, organizations, regions, or networks.

6.2.2. Hierarchical User-Groups Identification

Another form of user interaction analysis can be performed by grouping users into different clusters at different levels of granularity, based on the their tie strengths. User groups are generated at different hierarchies defined by the level of granularity. Hierarchical Agglomerative Clustering (HAC) is used to identify the hierarchical user-groups, which follows a bottom-up clustering approach to generate clusters containing sub-clusters and the process repeats to form a taxonomical structure [37]. It starts with considering every single object as a cluster, and each successive iteration agglomerates (merges) the closest pair of clusters based on some similarity criteria, until the agglomeration results into a single cluster. Given a set of n objects, the HAC algorithm aims to generate a sequence of nested disjoint partitions $C_1, C_2, \ldots, C_m$, where the number of partitions m at each level depends on the employed similarity function. The proposed method applies HAC on the constructed social graph G following a complete-link distance criterion, and the generated clusters are pictorially presented in Figure 8 as a bar chart. In this figure, there is a bar for each top-9 hierarchy levels and each kind of shades represent a cluster as defined by the legends. The figure highlights the portion of the total number of users lying in each cluster as the hierarchy level goes on increasing from top to bottom. Figure 9 shows the dendrogram representation for the generated results. It can be observed that at the top-most level of the hierarchy 456 users are grouped into two sub-clusters comprising 25 users and 431 users. At the next level the 431 users cluster is sub-grouped into 413 users and 18 users, and so on. Investigation through such user-groups has the potential to discover facts of taxonomical importance. Association of users can be analyzed starting in a broad perspective from the top and moving towards the bottom of the hierarchy for a closer and specific watch (or vice versa) at different levels of granularity.

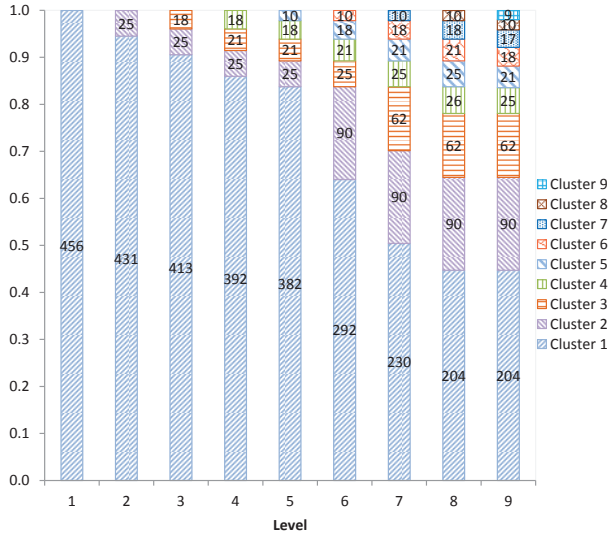
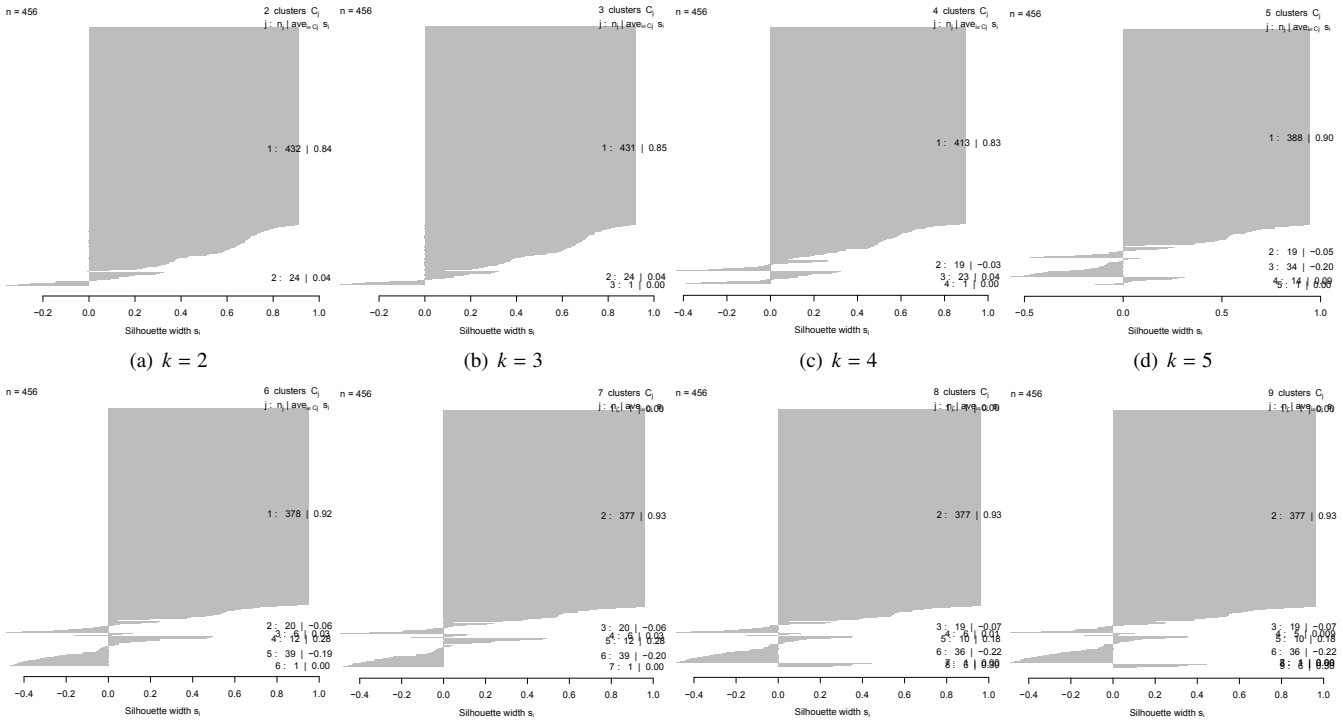


Figure 8: Hierarchical clusters

6.2.3. Random-Walk User-Groups Identification

A third way to analyze user interactions is through simulating a random walk from one user to another, where the move of walk is influenced by their social tie. The proposed method applies Markov CLustering (MCL) on the adjacency matrix A_G of the social graph G, and the generated clusters are pictorially presented in Figure 10 as a bar chart. There is a bar for each value of the inflation parameter starting at 1.4 to 2.1 at intervals



(a) $k = 2$ (b) $k = 3$ (c) $k = 4$ (d) $k = 5$

(e) $k = 6$ (f) $k = 7$ (g) $k = 8$ (h) $k = 9$

Figure 7: Partitive clustering results in terms of Silhouette measure

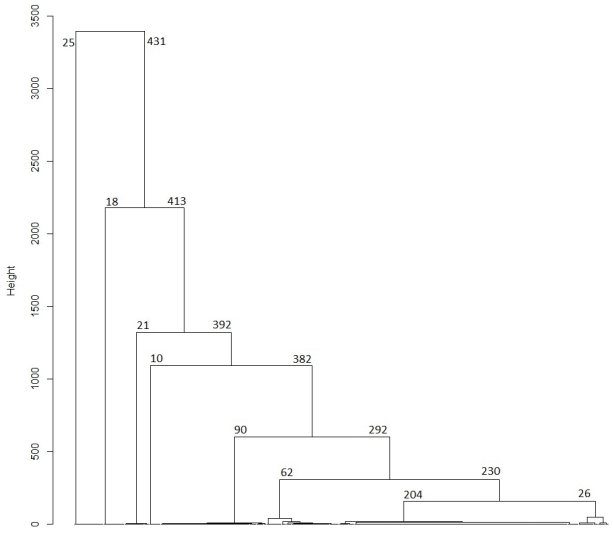


Figure 9: Hierarchical clustering dendrogram

of 0.1, and each kind of shades represents a cluster as defined by the legends. The figure highlights the portion of the total number of users lying in each cluster as the inflation parameter increases. Figure 11 presents some insights to the clustering results. Figure 11(a) shows the impact of the inflation parameter r on the number of iterations required to attain the convergence. The number of iterations required to converge at each value of r from 1.2 to 2.1 at intervals of 0.1 are 36, 22, 17, 22, 13, 14, 13, 11, 9, and 5 respectively. The line curve goes down from left to right, except at 1.5, which indicates that for the increasing values of the inflation parameter the number of iterations required for convergence decreases. Figure 11(b) shows the impact of the inflation parameter on the generated number of clusters. The number of generated clusters at each value of r from 1.2 to 2.1 at intervals of 0.1 are 24, 17, 11, 7, 3, 3, 2, 2, 2, and 1 respectively. The line curve goes down from left to right, which means that for the increasing values of the inflation parameter the number of generated clusters decreases. The importance of investigation through such user-groups lies in the scenario when there is absolutely no clue with the investigators about the participating users. By varying the inflation parameter, random-walk user-groups can be identified with varying levels of user similarity or dissimilarity in a group.

7. Conclusion

In this paper, we have presented a social graph based unified text mining framework to analyze chat logs' metadata and message contents in an integrated manner for cyber-crime investi-

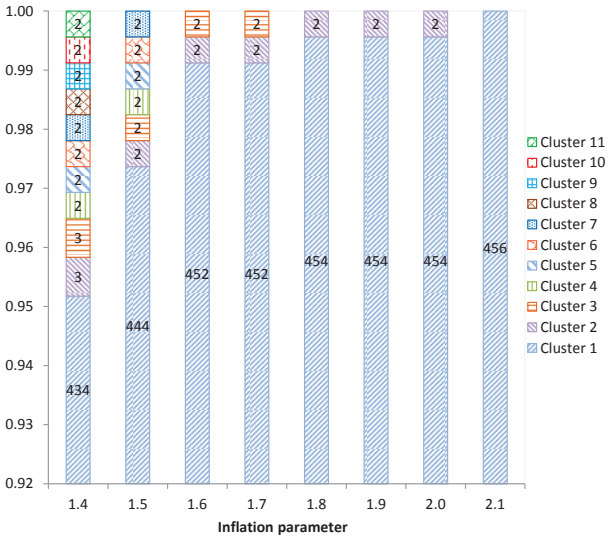
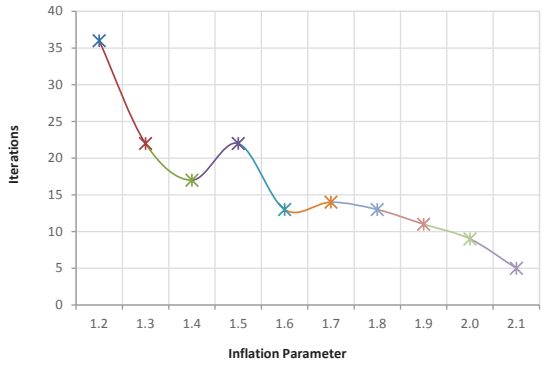
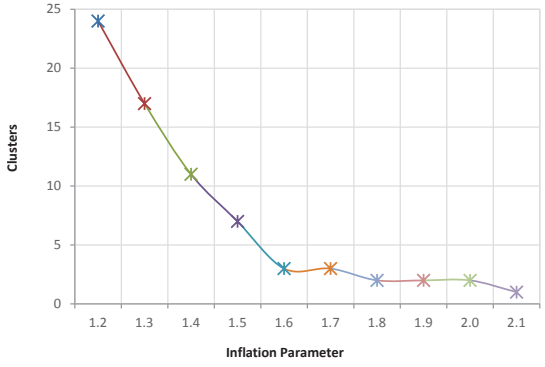


Figure 10: Random-walk clusters



(a) No. of iterations



(b) No. of clusters

Figure 11: Clustering results using MCL algorithm

gation. The metadata gives information about the user interactions, whereas the message contents give information about their textual conversations. The chat messages usually contain large amounts of noise and follows informal styles without caring much about spelling and grammatical correctness. The proposed framework employs a rich set of text pre-processing methods to normalize and smoothen the informal and noisy chat messages. The novelty of this paper lies in the social graph construction method that exploits both the metadata and the message contents to model users and their ties using a weighted graph. The relationships between different users are drawn by their participation in common group-chat sessions. Each user interest is represented by a set of key-terms extracted from the message contents using the n-gram technique and a customized HITS algorithm, and the strength of the tie between each pair of users is determined as a function of their overlapping interests. We have presented three possible crime investigation scenarios and proposed partitive, hierarchical, and random-walk user-group identification methods that are based on different clustering algorithms having specific properties. Thus the framework mines key-information as key-terms, key-users, and key-sessions, and different user-groups. For cyber-crime investigation, these information could lead to discover many critical facts and clues to trace the real culprits. The investigation can be further refined and strengthened by recovering more chat logs from computers of other users in the suspect's network, and getting results on the combined dataset.

References

[1] M. Abulaish, T. Anwar, A web content mining approach for tag cloud generation, in: Proc. of the 13th Int'l Conf. on IIWAS, 2011, pp. 52–59.

[2] M. Abulaish, T. Anwar, A keyphrase-based tag cloud generation framework to conceptualize textual data, Int'l J. of Adaptive, Resilient and Autonomic Systems (IJARAS) 4 (2) (2013) 72–93.

[3] P. H. Adams, C. H. Martell, Topic detection and extraction in chat, in: Proc. of the IEEE Int'l Conf. on Semantic Computing, ICSC '08, 2008, pp. 581–588.

[4] S. Agarwal, S. Godbole, D. Punjani, S. Roy, How much noise is too much: A study in automatic text classification, in: Proc. of the 2007 Seventh IEEE Int'l Conf. on Data Mining, 2007, pp. 3–12.

[5] R. Al-Zaidy, B. C. M. Fung, A. M. Youssef, F. Fortin, Mining criminal networks from unstructured text documents, Digital Investigation 8 (3-4) (2012) 147–160.

[6] J. Allan, C. Wade, A. Bolivar, Retrieval and novelty detection at the sentence level, in: Proc. of the ACM SIGIR conf., 2003, pp. 314–321.

[7] T. Anwar, M. Abulaish, Web content mining for alias identification: A first step towards suspect tracking, in: Proc. of the IEEE Int'l Conf. on ISI, 2011, pp. 195–197.

[8] T. Anwar, M. Abulaish, Identifying cliques in dark web forums- an agglomerative clustering approach, in: Proc. of the IEEE Int'l Conf. on ISI, 2012.

[9] T. Anwar, M. Abulaish, Namesake alias mining on the web and its role towards suspect tracking, Information Sciences 276 (2014) 123–145.

[10] T. A. Aw, L. H. Lee, Personalized normalization for a multilingual chat system, in: Proc. of the 50th Ann. Meeting of the ACL, 2012, pp. 31–36.

[11] N. L. Beebe, J. G. Clark, Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results, Digital Investigation 4 (Supplement) (2007) 49–54.

[12] J. Bengel, S. Gauch, E. Mittur, R. Vijyaraghavan, Chattrack: Chat room topic detection using classification, in: Proc. of the 2nd Symposium on ISI, 2004, pp. 266–277.

[13] D. Bogdanova, P. Rosso, T. Solorio, On the impact of sentiment and emotion based features in detecting online sexual predators, in: Proc. of the

ACL 2012 Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, 2012, pp. 110–118.
[14] H.-C. Chu, D.-J. Deng, J. H. Park, Live data mining concerning social networking forensics based on a facebook session through aggregation of social data, IEEE Journal on Selected Areas in Communications 29 (7).
[15] M. Elsner, E. Charniak, Disentangling chat, Comput. Linguist. 36 (3) (2010) 389–409.
[16] T. Holmer, Discourse structure analysis of chat communication, Language@Internet 5.
[17] F. Iqbal, B. C. M. Fung, M. Debbabi, Mining criminal networks from chat log, in: Proc. of the IEEE/WIC/ACM Int'l Jt. Conf. on Web Intelligence and Intelligent Agent Technology, WI-IAT '12, 2012, pp. 332–337.
[18] A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.
[19] A. Kamal, M. Abulaish, T. Anwar, Mining feature-opinion pairs and their reliability scores from web opinion sources, in: Proc. of the 2nd Int'l Conf. on WIMS, 2012.
[20] J. M. Kleinberg, Authoritative sources in a hyperlinked environment, J. ACM 46 (5) (1999) 604–632.
[21] T. Kucukyilmaz, B. B. Cambazoglu, C. Aykanat, F. Can, Chat mining: Predicting user and message attributes in computer-mediated communications, Information Processing and Management 44 (4) (2008) 1448–1466.
[22] Y. Lee, H.-y. Jung, W. Song, J.-H. Lee, Mining the blogosphere for top news stories identification, in: Proc. of the 33rd Int'l ACM SIGIR Conf., 2010, pp. 395–402.
[23] A. Louis, A. Engelbrecht, Unsupervised discovery of relations for analysis of textual data, Digital Investigation 7 (3–4) (2011) 154–171.
[24] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proc. of the 5th Berkeley Symp. on Math. Statist. and Prob., vol. 1, 1967, pp. 281–297.
[25] B. O'Connor, R. Balasubramanyan, B. R. Routledge, N. A. Smith, From tweets to polls: Linking text sentiment to public opinion time series, in: Proc. of the AAAI ICWSM Conf., 2010.
[26] O. Özyurt, C. Köse, Chat mining: Automatically determination of chat conversations' topic in turkish text based chat mediums, Expert Systems with Applications 37 (12) (2010) 8705–8710.
[27] M. F. Porter, An algorithm for suffix stripping, Program: electronic library and information systems 14 (3) (1980) 130–137.
[28] A. P. Schmidt, T. K. M. Stone, Detection of topic change in irc chat logs, http://www.trevorstone.org/school/ircsegmentation.pdf.
[29] D. Shen, Q. Yang, J.-T. Sun, Z. Chen, Thread detection in dynamic text message streams, in: Proc. of the 29th Ann. Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval, SIGIR '06, 2006, pp. 35–42.
[30] I. Shklovski, L. Palen, J. Sutton, Finding community through information and communication technology in disaster response, in: Proc. of the 2008 ACM Conf. on Computer Supported Cooperative Work, 2008, pp. 127–136.
[31] K. Spärck Jones, Automatic summarising: The state of the art, Inf. Process. Manage. 43 (6) (2007) 1449–1481.
[32] D. P. Twitchell, N. Forsgren, K. Wiers, J. K. Burgoon, J. F. Nunmaker, Jr., Detecting deception in synchronous computer-mediated communication using speech act profiling, in: Proc. of the 2005 IEEE Int'l Conf. on Intelligence and Security Informatics, ISI'05, 2005, pp. 471–478.
[33] D. C. Uthus, D. W. Aha, Multiparticipant chat analysis: A survey, Artificial Intelligence 199-200 (2013) 106–121.
[34] S. Van Dongen, A cluster algorithm for graphs, Ph.D. thesis, University of Utrecht (2000).
[35] J. B. Walther, K. P. D'addario, The impacts of emoticons on message interpretation in computer-mediated communication, Social Science Computer Review 19 (3) (2001) 324–347.
[36] Y. Xiao, J. Yu, Partitive clustering (k-means family), Wiley Int. Rev. Data Min. and Knowl. Disc. 2 (3) (2012) 209–225.
[37] M. J. Zaki, W. Meira Jr., Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, 2014 (to appear).