A Framework for Mining Instant Messaging Services

John Resig   Ankur Teredesai

Data Mining Research Group
Department of Computer Science
Rochester Institute of Technology
{jer5513,amt}@cs.rit.edu

Abstract

Developing a framework for analysis of large scale mass-communication media such as instant messaging (popularly known as IM) has gone largely unexplored up until this point. This paper explores various data mining issues and how they relate to Instant Messaging and current Counter-Terrorism efforts. Specific topics include user pattern analysis, anomaly detection, limited message size based textual topic detection, and largely generic social network analysis in this context. Several interesting questions are posed and the current framework being developed explores some of the possible solutions.

1 Introduction

The medium of Instant Messaging on the Internet is a well-established means by which users can quickly and effectively communicate with one another. Long utilized by the public as a quick form of free communication, data mining tasks have not been attempted over Instant Messaging. Additionally, on a corporate or government level, people are just beginning to take notice of the potential that IM provides in terms of the type of information that can be collected from these networks. Many large software or internet based corporations have started Instant Messaging networks of their own, generally open to the public after registration, including Time Warner, Yahoo, and Microsoft. Currently, some of the most popular Instant Messaging networks are run by some of the aforementioned companies:

- AOL Instant Messenger
- Yahoo! Instant Messenger
- MSN Instant Messenger
- Various IRC Networks

Interestingly enough, even with all the various networks being developed by corporations for profit, their physical structures (client-server architecture) and communication protocols (information packets) are very similar to one another.

Online	The user's client is connected to the central server and the user is active (currently typing or moving the mouse on his computer).
Offline	The user's client is not connected to the messaging server at this time.
Idle	The user's client is connected to the central server, but the user is not active. Additionally, how long a user has been idle can be determined from their status.
Away	The user is logged on but away from the station. Sometime users specify a text message that can be viewed by anyone who wishes to get more information about where they are or why they are away. (e.g. ”Out to lunch.”, ”Watching TV.”) In fact a user can be either idle, or active, while an away message is explicitly up.

Table 1: Possible user statuses. As shown above an IM client can be in one of the above statuses at a given time.

Most Instant Messaging networks follow a strict Client-Server model in which a server (or a cluster of servers) is maintained by a service provider who controls traffic coming to and from the server. Users who wish to utilize a certain network generally register themselves with the service provider, then download a provider-approved client for use on their network. Using this client, users can connect to the central server in order to be able to send and receive messages and collect account information. A *friend* is generally another registered user (the term *friend* is server-specific, but exists on almost all messaging networks). The concept is that a user may maintain a *Buddy List* under which a listing of their immediate friends may

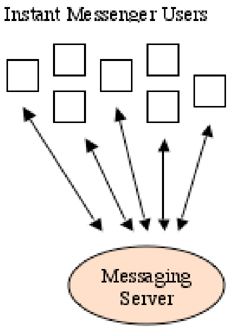

Figure 1: Existing Instant Messaging Network

exist. Using this, the server then sends a client updates based upon the statuses of their friends. Once the connection process has completed, the server performs all future communication in the form of *Update Packets*. An update packet is sent from the server to a client whenever an action occurs that is associated with him. For example, when a friend performs a status change or if a message is being sent to a user's client. An unfortunate consequence of the server maintaining such buddy lists is that it can impose restrictions upon the maximum number of friends which a user is allowed to maintain (this number is generally around 200). Since a client does not directly communicate with any other connected client, and only the server, the server is then in charge of disseminating any potentially useful information from one client to another. Once such piece of critical information is a user's status. Table 1 describes a list of possible statuses that a client can be in. Status is an attribute generally associated with a user's client and often indicates how a user responds to an Instant Message. Whenever a user's status changes, an update packet is relayed by the central server to everyone who has the user on their buddy list.

Another important aspect of communication flow within an Instant Messaging network is the traffic of messages between users. The amount of information revealed concerning instant messages is generally limited to the information which is directly related to a user. Such information paths include chat rooms (a group discussion area where multiple people can communicate with one another stimultaneously) and private Instant Messages (messages sent directly from one user to another).

2 Data Collection

Between the various information resources provided by Instant Messaging networks, there are a number of valuable resources available to the average user. The

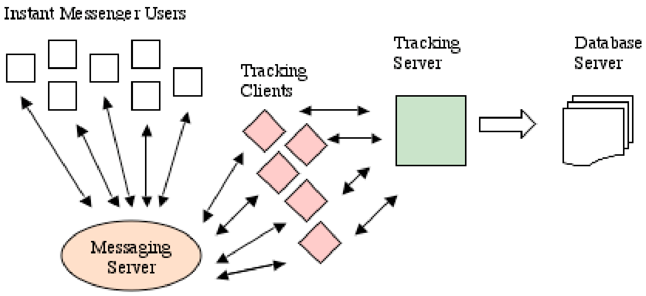

Figure 2: The Proposed IM Mining Framework

data generated in turn is very useful for data mining to analyze user behavior. However, in order to utilize the flow of information offered by these networks, a data collection framework need will have to be established. This paper proposes one such framework which has been developed. Information distributed by the Instant Messaging networks can be broken down into two simple groups: User status-change and communication-flow (Instant Messages, Chat Rooms).

The first item collected, user status change, can be achieved relatively simply as the current structure of Instant Messaging networks support the collection process. One interesting feature, previously discussed, of Instant Messaging networks is that of 'Buddy Lists' - lists of friends of a user. The direct benefit of this feature is the fact that whenever a buddy (a member of a user's buddy list) performs a status change, the client is immediately notified of it by the server. Utilizing this feature set, one could set up a client of their own, with an arbitrary buddy list, and begin collecting information about their 'buddies' resulting actions. This is significant due to the fact that most Instant Messaging networks don't require that someone actually be a friend of another user in order to watch their status changes.

Using this standard model, it is relatively simple to set up a tracking client whose only job is to collect pertinent information about users that are on its buddy list - aptly named, in this framework, *Tracking Client*. In order to maintain a tracking client a *Tracking Server* is constructed which manages the actions of its associated tracking clients. The *Tracking Server* marshalls communication between an arbitrary number of tracking clients and the database server. Whenever a new *Tracking Client* spawns and connects to the *Tracking Server* the server attempts to determine which Instant Messaging users need to be tracked, from a list of potential users. Due to the restrictions imposed by the various Instant Messaging networks as to the size of a user's buddy list this distributed *Tracking Client* structure is required in order to be able to track the maximum number of people at any given time. An advantage to this distributed network is that no one client is dependant upon for all tracking efforts or network bandwidth usage. Each *Tracking Client* within the network watches a given number of other clients in order to verify that they are, in fact, still connected to the network - if not then a communication is sent to the tracking server and another client is spawned to cover the users not being tracked by its disabled peer. As information packets come in from the server to each tracking client, the client attempts to determine if the packet should be re-transmitted to the server for storage in the central database.

Another tracking effort that is currently being explored is that of monitoring inter-user communication. One resource offered by most Instant Messages networks (and exclusively by others, see IRC) is that of a public chat room. A tracking client has the ability to connect to one of these rooms as a spectator, simply to view the flow of conversation. Similar to how the server performed by sending data packets concerning a user's status change, the server will also send packets detailing messages being publicly sent from one user to another within this chat room setting. As with status changes these packets are verified for integrity and then passed along to the tracking server for subsequent storage. Packet integrity is verified by checking the information against the previously collected packets, making sure that no duplicate packets are transmitted to the server.

A new favorable advancement has recently been made by the AOL Instant Messaging network to allow a user to connect to the network from multiple locations using multiple clients. Using this pseudo-proxy, AOL displays only the users most-active (The order of activeness being: Online, Idle, Away, Away and Idle)connection to other users of the network. However, clients at equal states of activity receive all incoming communications. This advancement is very important due to the fact that now it is possible to spawn tracking clients for willing users of the network and provide them additional intelligent services on top of their normal Instant Messaging experience. It is expected that other Instant Messaging services will soon follow suit with a similar feature - to which additional services can then be provided to the users of those networks.

3 Collection Results

Using the previously discussed framework, some basic data collection of user status changes have been performed. The initial data set, shown in Table 2, was collected over a time period of 67 days during the summer of 2003. 207 participants were tracked and 55061 unique data packets were received and stored. Two separate tracking clients were used to collect results, both of which aggregated their data to a single database for later retrieval.

Figure 3 shows the probability that a given user was in a certain state over the course of 10 weeks. It can be quickly surmised that most users have the ability to maintain a fairly steady record from week-to-week. Additionally, due to the polar differences that some of the users seem to exhibit, it becomes apparent that the concept of *User Profiles* is an important step to determining a user's common course of action (more information can be found in Section 5.1 *User Pattern Analysis*).

Timestamp	User ID	Status
70242	68	Online
70303	118	Online
70325	65	Offline
70447	68	Idle
70453	16	Idle
70725	98	Offline
70743	89	Away
70824	98	Idle
70853	77	Offline
70978	120	Online
71006	120	Away

Table 2: Tracking Results: Sample Data

4 Data Processing and Storage

Another important portion of the data collection framework is the method by which the collected data will be stored for later interpretion. Currently, a database server is being used to store information as it filters in from the tracking clients. The biggest issue at hand is that of the volume of data being generated by the real-time stream of information coming off the network. In order to lower the potential storage requirements, and possibly even limit the amount of computation needed to be performed later, the framework employs data stream algorithms [27, 8] for data management.

The algorithmic improvements discussed in the StatStream [27] paper are applicable to the current restraints presented the Instant Messaging infrastructure. It is possible that anywhere from 10,000 to a few million users could be tracked using this framework (mainly due to the overwelming parallelism offered by this algorithm). While an algorithm of this nature may not be able to reduce the size of the data set directly it does offer the unique ability to quickly find correlations between similar streams in a pseudo real-time nature (this becomes more important in the *Anomaly Detection* stage of analysis).

The second algorithm for clustering data streams [8] can be adapted to aid in the efficient interpretation of user actions and the storage thereof. Using the clustering technique described in the paper and providing a limited number of clusters into which data points must be grouped, incoming data can be quickly generalized, requiring less data to be kept both in memory and in the database. By using the clustering algorithm to determine a common profile for a user, individual status changes will not need to be stored in the database, simply requiring the storage of much smaller culmulative profiles. The algorithm generates a number of median numbers upon which all future numbers can be approximated. Using this method, and a relatively small number of medians, say 24 hours in a day or even 48 half-hours, it can quickly aid in future *User Pattern Analysis* for a generalized period of time, thereby requiring less data to be stored by the application and less on-the-spot processing in the future.

5 Counter-Terrorism and Data Mining

Since September 11, 2001, and the increased terrorist activity against the United States, the area of Counter-Terrorist Data Mining has seen a surge of interest and papers relating to applications of old data mining techniques to a new field of study. Most papers attempt to utilize the study of Social Network Analysis in order to find potential links between suspicious groups of people. Two such works include the *Mapping Networks of Terrorist Cells* [14] and *A CBR Approach to Assymetric Plan Detection* [5].

The first paper concerning Counter-Terrorist activity is that of *Mapping Networks of Terrorist Cells* [14] and directly concerns the social network analysis surrounding the Sept. 11 attacks. This paper attempts to reveal the difficulty in attempting to find direct correlating activity between a sparsely related group of people. Mentioned in this paper is the tracking of *Task* Data Sources, which includes the use of Chat Rooms and Instant Messaging.

The paper '*A CBR Approach to Asymmetric Plan Detection*' [5] attempts to coordinate a social network of people and places with links of seemingly trivial actions. The concerns of this work surround three major issues: Massive data sets, noise, and incomplete information. In our framework we deal with similar issues:

- Develop algorithms for handling streams of data as it collects to perform clustering and averaging of

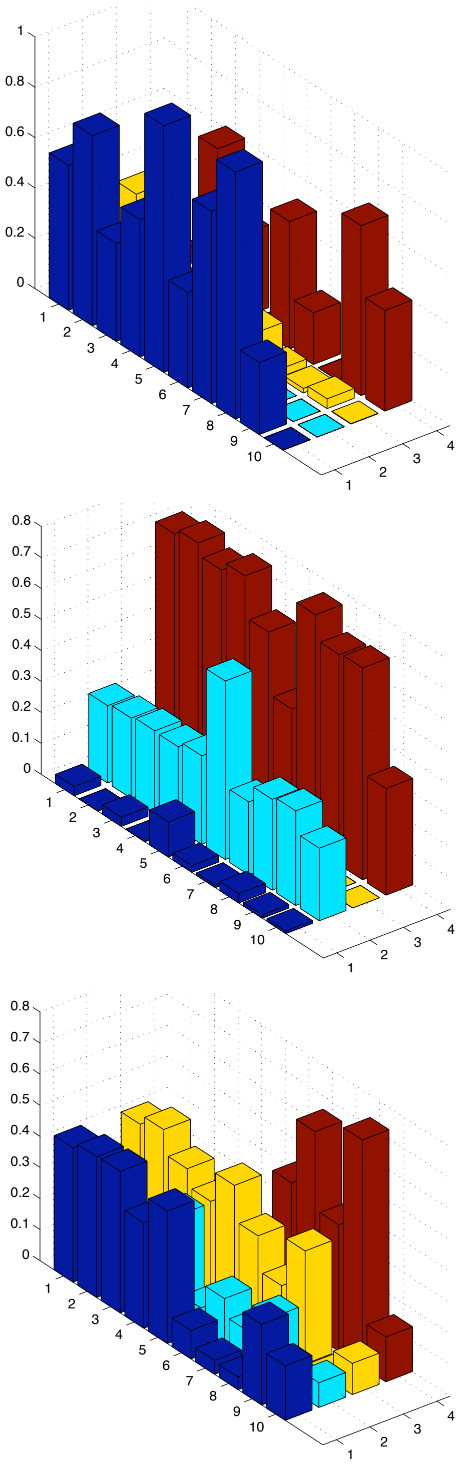

Figure 3: Tracking Results: Users 6-8 X-Axis: Weeks 1-10, Y-Axis: Probability that a user was a given status during that week, Z-Axis: Possible User Status (1 = Offline, 2 = Away, 3 = Idle, 4 = Online)

the information being transmitted.

- Data noise is filtering through the use of textual topic detection, attempting to determine common overall themes by flagging them for further cluster analysis.

- Incomplete information is minimized through additional analysis of data collected from social networks and textual analysis proceedures. This process is completed by coordinating actions with clusters of people.

The research topics presented by current counter-terrorist research represent the challenges that exist within this field. By adapting the algorithmic techniques (such as *Social Network Analysis* and *Textual Topic Detection*) presented as solutions, new knowledge can be derived from the current information present in Instant Messaging networks.

5.1 User Pattern Analysis The first step, and potentially the most revealing, of the mining processes applied to the information accrued by the framework is that of user pattern analysis. The overall goal of the pattern analysis is to attempt to construct an accurate profile which conveys information concerning a user's Instant Messaging usage strictly from information which is publically transmitted.

A lightweight version of the pattern analysis was presented in the Nov 2003 issue of the ACM Queue magazine [21], entitled 'Beyond Instant Messaging'. Within this article a concept called rhythm awareness was discussed, which is, essentially, the mapping of user activity over a period of time, attempting to determine what patterns a user followed and, subsequently, what actions a user is most likely to take at any given time. They constructed this into a mock-up under which a user is shown along with an associated time at which they will come back online (or go away, depending on the user). Something similar to this could be constructed on a user level however on a large-scale tracking level it becomes much more apparent that robust user profiles need to be quickly constructed based upon as little information as possible, as quickly as possible.

The current effort of this project is to attempt to use various clustering algorithms [8, 26, 9] to build an accurate profile of when users are most likely to commit a certain action. Depending the number of data points within the data set the granularity of clustering will differ - larger data sets may utilize a 24 cluster scheme under which user actions are attempted to fit within the hours of any given day. Using this clustering method it can be determined, with a certain level of confidence, that a user will perform a certain action during a certain time period. On this information analysis will be performed given a certain user and a given time period returning a profile of that user detailing his or her frequent patterns as they pertain to the use of Instant Messaging.

Previous work into the area of modeling user activities [6, 10, 22, 11, 25, 3, 23] has proved to be rather successful. Almost all of the previous methods attempt to use statistical analysis as the sole way to determine a user's next action. Other applications hook directly into known user schedules in an attempt to bypass any unintelligent determinations. Unfortunately, due to the open aspect of Instant Messaging networks such a liberty is not available. In order to remove any false schedules potentially generated by the analysis tools the network will be under a constant state of training and re-clustering in an attempt to absolutely verify user's actual patterns.

One interesting method [18] presented is that of finding *Generalized Episodes Using Minimal Occurrences*. The simplicity of the algorithms offered in the paper lend it to being used on a user-by-user basis for quickly determining their next possible action. This novel concept could be offered as a quick alternative to users who wish to find out more information about their buddy list and do not want to wait for any culmulative results from a clustering engine.

5.2 Anomaly Detection One aspect of this paper that elevates tracking Instant Messaging from simple pattern analysis to the level of power and complexity expected when utilizing a Counter-Terrorism application is that of anomaly detection. The area of anomaly cetection is well-researched and offers many previous applications on which to gain an excellent basis for study. Perhaps the most prominently used example of the topic is that of *Intrusion Detection Systems* [16, 15, 17, 2, 20]. These detection systems are generally placed on computer systems or networks and are tasked with collecting information on all the actions of its users. Based upon the collected information it will be determined whether or not a user is performing abnormal activity. Some systems even go so far as to detect the activity and immediately block any future actions by the user - without the need for human intervention. Due to the network structure of most Instant Messaging services, the model of intrusion detection is prefectly applicable and beneficial.

Discussed in some of the papers [2, 20] is the concept of Profile-Based Anomaly Detection. This idea is central to the detection process which will be employed.

From the results of the *User Pattern Analysis*, profiles will have been constructed by clustering common network users together into similar groups. Using these common profiles (and taking into account the potential for outliers) they will be used to quickly scan against those who had previously fit them and immediately flag those who derivate from that trend. In order to compensate for a miscommunication of urgency (e.g. somebody goes on vacation for a weekend) each profile will be built to include acceptable derivations which a user can enjoy without calling undeserved attention to themselves. These flexible profile-clusters will be essential to accurately determining a user's true actions.

In order to build an accurate user profile upon which to compare incoming results, it becomes apparent that having a single profile-building method would not be an accurate method for determining a user's next action. There are three sub-profiles that will be built and subsequently compared in order to acheive the best results:

- Compare data against a user's previous actions. By constructing a single profile of a user's activity patterns it can be used for quick comparisions to determine when a user strays from their personal patterns.

- Compare data against the actions of people close to a user. By constructing an aggregate profile which consists of strictly close friends who maintain similar patterns of activity. Using this method, trends within a group of friends can be detected and dealt with accordingly.

- Compare data against larger community. A large community of friends would consist of buddy relations multiple levels deep. Using this method it would be possible to spot or normalize trends within an entire community or locale.

Using these profiles together will provide a deeper level of understanding and detection that were not previously available by using a simple single layer of analysis. This additional form of abnormal behavior being detected will greatly benefit Counter-Terrorism efforts who wish to monitor certain users for out of the ordinary behvior patterns.

5.3 Social Network Analysis A powerful area of analysis that is frequent to most areas of communication is that of Social Network Analysis [19, 24] and Instant Messaging is no exception. There are a few aspects that are perfect for various link analysis techniques:

- Buddy Lists. Buddy Lists contain a list of immediate *friends* which can be immediately translated over to *peers* within a social network.

- Private Messages. These provide excellent weights for which a social network can be structured.

- Public Chat Rooms. These locations are an excellent example of group social networks within which users can communicate and associate with each, constructing a complete ad-hoc network.

A user's personal Buddy List is an excellent building block for constructing further social networks. A unique aspect of the network that is created by the list is its completely ad-hoc nature under which the owner has the ability to be add or remove users completely at random. Using this small group, various user profiles can be created. This group can help give some contextual aide to planning and *Anomaly Detection* within a network. Additionally, by using a network that is a couple links deep, a regional network will begin to take shape, thusly helping in regional pattern analysis.

The importance of tracking the frequency of Instant Messages cannot be overstated. By tracking how often a user communicates with his peers, a link weighting [7] schme can be immediately applied to the previously-constructed social network. Normally, a basic network would have no detail about how 'good' of a friend a peers are. By using the frequency of Private Instant Message communication as a weighting metric an inter-user relationship can be quickly surmised for strength. By adding weight to the network scheme, social groups begin to come to light: People who tend to only communicate with each other. All of this is very important to properly and accurately conveying how an Instant Messaging network is socially constructed.

A final interesting aspect of social networks is as to how they apply to Public Chat Rooms. The structure of the *chat room* is very different from the traditional network, as presented before. This construction is far more ad-hoc in nature - chat participants come and go as they please, frequently having no real connection to the discussions or people doing the discussing. By using the, sometimes brief, encounters as weak links between users you can begin to see social groups of people who have similar interests. As with Private Instant Messages, as the frequency of discussion by a user is increased (linking this in with *Textual Topic Detection*) concerning a certain topic, that information can be attributed to them. Another interesting aspect of the chat room setting is watching for the spread of news and other topics of interest, as the information migrates from room to room, or from user to user.

Social networks are an important part of Counter-Terrorism efforts [5, **?**] as they are frequently used when trying to determine plans before they become too late and who people tend to collaborate with. Most importantly, the efficient tracking and analysis of relationships through seemingly trivial discussion within a chat room could lead to new groups that were not previously apparent.

5.4 Textual Topic Detection Within the spectrum of analysis offered by the previous pattern analysis, additional information is available which could aide the previously-detected results. Three pieces of textual information are provided by a user and are available for further mining which could aide in both more accurate pattern detection and a whole new layer of social network analysis. The three areas of textual information are as follows:

- Away message text. Whenever users go into an 'Away' status an associated text message is displayed (specified by the owner) to all those who are interested. Typically the message contains information that is pertinent to the user's current activity or location.
- Private Message text. Due the direct nature of inter-user messages topics discussed will gain a greater weight with their owners, it can generally be associated that topics discussed within these conversations have direct implications with those participating.
- Public Chat Room text. As chat room participation is mined for link analysis between multiple users, so are the textual references mined for associated topics - a powerful additional layer of analysis to which additional meaning can be given to group actions.

The premise of the Away Message is both interesting and complex as it offers a complex new variable to the process of *User Pattern Analysis*. When a user goes into an away status, he is providing additional information pertaining to his whereabouts and current activities. By analyzing this information, and by associating it with common activities, it may become possible to create a more accurate picture of a user's activity pattern. The issue at hand with Away Messages, however, is that messages are sometimes devoid of relevant information, misleading, or simply don't contain enough information to be deemed relevant. In order to counter this, the seemingly most useful way to mine away messages is by looking for relevant keywords which are associated with activities (e.g. Lunch, Dinner, School, Work, etc.). Based upon the presence of these keywords a given away message could be flagged as being able to provide further inforamtion to user pattern analysis.

Private Instant Messages provide significant textual resources for additional contextual information. By using a level of topic detection [4, 1] across all conversations new links will begin to appear that may not have existed previously. For example: If a user A talks with user B about a specific topic and then turns around and messages user C about the same topic, we can assume that there is a certain degree of connection between B and C that was not seen before (whether it be a physical or simply a connection based upon similar interests).

The final area of improvement encompesses Chat room topic detection [13, 12]. A community-based discussion atmosphere, in which group communication happens frequently, provides additional information then just user-to-user discussions. Topic detection within this group atmosphere is very important to determining what trends are occuring within a larger set of people. By determining what a set group of people are discussing, it could be seen how well messages are transmitted from one person to another group of people.

6 Conclusion

The data being collected from Instant Messaging networks brings exciting new applications to existing research topics within Data Mining. Using the techniques of User Pattern Analysis, Anomaly Detection, Textual Topic Detection and Social Network Analysis new advancements can be made to current Counter-Terrorism efforts by bringing the addition of a new, trackable, data medium that has previously gone largely unused. In this paper we have described a data collection framework to support these initiatives. We have started developing the analysis components and initial results will be available by March. The central idea is to present this framework and obtain feedback regarding this initiative.

References

[1] Allan, J., Carbonell, J., Doddington, G., Yamron, J., and Yang, Y. Topic detection and tracking pilot study: Final report, 1998.

[2] Anderson, D., Frivold, T., Tamaru, A., and Valdes, A. Next-generation intrusion detection expert system (nides), software users manual, beta-update release. Tech. Rep. SRI–CSL–95–07, Computer Science Laboratory, SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025-3493, May 1994.

[3] Begole, J. B., Tang, J. C., Smith, R. B., and Yankelovich, N. Work rhythms: analyzing visualizations of awareness histories of distributed groups. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work* (2002), ACM Press, pp. 334–343.

[4] Bingham, E., Kabn, A., and Girolami, M. Topic identification in dynamical text by complexity pursuit.

[5] Daniel Fu, E. R., and Eilbert, J. A cbr approach to asymmetric plan detection. In *Proceedings of the Workshop on Link Analysis for Detecting Complex Behavior (LinkKDD2003)* (2003), ACM Press.

[6] E. Horvitz, P. Koch, C. K., and Jacobs, A. Coordinate: Probabilistic forecasting of presence and availability. In *Proceedings of the 2002 Conference on Uncertainty and Artificial Intelligence* (2002), AAAI Press, pp. 224–233.

[7] Graves, T. L. Finding clusters in network link strength data, 1998.

[8] Guha, S., Mishra, N., Motwani, R., and O'Callaghan, L. Clustering data streams. In *IEEE Symposium on Foundations of Computer Science* (2000), pp. 359–366.

[9] Guha, S., Rastogi, R., and Shim, K. CURE: an efficient clustering algorithm for large databases. pp. 73–84.

[10] Hill, R., and Begole, J. B. Activity rhythm detection and modeling.

[11] Hudson, S. E., Fogarty, J., Atkeson, C. G., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J. C., and Yang, J. Predicting human interruptibility with sensors: A wizard of oz feasibility study.

[12] Khan, F. M., Fisher, T. A., Shuler, L., Wu, T., and Pottenger, W. M. Mining chat-room conversations for social and semantic interactions.

[13] Kolenda, T., Hansen, L., and Larsen, J. Signal detection using ica: application to chat room topic spotting, 2001.

[14] Krebs, V. Mapping networks of terrorist cells, 2002.

[15] Kumar, S. *Classification and Detection of Computer Intrusions.* PhD thesis, Purdue, IN, 1995.

[16] Lee, W., and Stolfo, S. Data mining approaches for intrusion detection. In *Proceedings of the 7th USENIX Security Symposium* (San Antonio, TX, 1998).

[17] Lunt, T. F. Detecting Intruders in Computer Systems. In *Proceedings of the Sixth Annual Symposium and Technical Displays on Physical and Electronic Security* (1990).

[18] Mannila, H., and Toivonen, H. Discovering generalized episodes using minimal occurrences. In *Knowledge Discovery and Data Mining* (1996), pp. 146–151.

[19] Newman, M., Watts, D., and Strogatz, S. Random graph models of social networks.

[20] Porras, P. A., and Neumann, P. G. EMERALD: Event monitoring enabling responses to anomalous live disturbances. In *Proc. 20th NIST-NCSC National Information Systems Security Conference* (1997), pp. 353–365.

[21] Tang, J. C., and Begole, J. B. Beyond instant messaging. 28–37.

[22] Tang, J.C., Y. N. B. J. V. K. M. L. F., and Bhalodia, J. Connexus to awarenex: Extending awareness to mobile users. In *In Proceedings of Conference on Human Factors in Computing Systems CHI '01* (2001), ACM Press, pp. 221–228.

[23] Tyler, J. R., and Tang, J. C. When can i expect an email response? a study of rhythms in email usage.

[24] Wasserman, S., and Faust, K. *Social Network Analysis: Methods and Applications.* Cambridge University Press, 1994.

[25] Zerubavel, E. *Hidden Rhythms: Schedules and Calendars in Social Life.* University of Chicago Press, 1981.

[26] Zhang, T., Ramakrishnan, R., and Livny, M. BIRCH: an efficient data clustering method for very large databases. pp. 103–114.

[27] Zhu, Y., and Shasha, D. Statstream: Statistical monitoring of thousands of data streams in real time, 2002.