Identifying Multi-ID Users in Open Forums*

Hung-Ching Chen, Mark Goldberg, and Malik Magdon-Ismail

Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180, USA.
Email: {chenh3,goldberg,magdon}@cs.rpi.edu, wallaw@rpi.edu.

Abstract. We describe a model for real-time communication exchange in public forums, such as newsgroups and chatrooms, and use this model to develop an efficient algorithm which identifies the users that post their messages under different IDs, *multi-ID users*. Our simulations show that, under the model's assumptions, the identification of multi-ID users is highly effective, with false positive and false negative rates of about 0.1% in the worst case.

1 Introduction

It is well known that some users, or *actors*, in communication networks, such as newsgroups and chatrooms, post messages using different actor IDs. We denote such actors as *multi-ID actors* or *multi-ID users*. As a rule, these actors attempt to hide the fact that one person is operating using multiple IDs. The reasons for an actor to use multiple IDs are varied. Sometimes, an actor who has become a pariah of a certain public forum may try to regain his/her status using a different ID; sometimes, Actors may post messages under different IDs in order to instigate debate or controversy; sometimes, actors may pose as multiple IDs in order to sway democratic processes in their favor in certain voting procedures on Internet forums, such as a leadership election. In general, the identification of an actor who posts under several IDs should have important forensic value. At a very least, flagging IDs as possibly belonging to the same actor of a public forum can justify further investigation of the posts under those IDs.

A variety of approaches could be adopted to identifying multi-ID actors. Though tracing the source of an Internet packet is not trivial, it is technically possible to identify the IP address of the packets sent by different IDs. While this certainly help with the identification, this information is often insufficient, since a single IP address could represent a cluster of computers on a local network. Furthermore, access to those computers could be available to different people, as is the case for computer laboratories in universities. Another approach would be to analyze the semantics of the messages. For example, a particular actor may use a particular phrase in all of the IDs that it operates. Semantic analysis would attempt to discover stylistic similarities between posts of the same actor using different IDs. This entails sophisticated linguistic analysis that is generally not efficient, and may not be easy to automate, which becomes a serious obstacle if

* This research was partially supported by NSF grants 0324947 and 0346341

the task is to identify multi-ID users operating in several out of potentially many thousands of communication networks. Below is a sample log from a chatroom, which illustrates the difficulty of any form of linguistic analysis.

[20 : 01 : 18] <id_1> I shall powerful fart from apple pie, than from hamburger
[20 : 01 : 41] <id_2> some girls who want to chat with a male with webcam??
[20 : 01 : 55] <id_3> I shall powerful fart from mom beans than from american taco
[20 : 01 : 57] <id_3> hahahahahahahahah
[20 : 02 : 18] <id_4> hey <BB> id_{11} <AB> : i'm happy :P
[20 : 03 : 35] <id_1> Big farting from salad Olivier!
[20 : 03 : 40] <id_5-> be nice id_4
[20 : 03 : 47] <id_5-> or be gone
[20 : 04 : 18] <id_7> still searching for guys between 35 nd 45
[20 : 04 : 23] <id_4> <BB> id_5- <AB> : did i talked to yu
[20 : 04 : 26] <id_4> did i say somthing bad
[20 : 04 : 30] <id_8> id_9@hotmail.com
[20 : 04 : 30] <id_8> id_9@hotmail.com
[20 : 04 : 30] <id_8> id_9@hotmail.com
[20 : 04 : 40] <id_4> she know what she do if she want to kick me she will
[20 : 04 : 49] <id_{10}> hello any hot girl that wanna chat pick me

In this paper, we present an altogether different approach, based upon statistical properties of the posts. To take the chatroom as an example, each post has three tags associated with it, <t,id,message>: t is the time of the post; id is the ID posting the message; and message is the message that was posted. The question we address is whether it is possible to identify the multi-ID users based only on the times and IDs of the posts, *i.e.*, ignoring the actual message texts.

We describe a model that provides a realistic emulation of a live public forum, such as a chatroom or a newsgroup. This model is based on viewing each actor as a queue of "posts-in-waiting." Based upon the messages that are delivered by the server and the list of its "friends", every actor builds up its queue of jobs–the replies to messages received by the actor. The actor processes each of these messages one by one and submits each reply to the server; these messages then generate reply-jobs in queues of other IDs (the friends), and so on. Using such a model as a foundation, we discover statistical parameters that differentiate between multi-ID actors and single-ID actors. These parameters are the result of numeric and combinatorial analysis of the sequence of posts which (the analysis) does not use semantic information regarding the texts of the posts. The main observation, which forms the basis of our algorithm, is that the pots tagged with an ID operated by a multi-ID actor do not appear as frequently as do the posts of single-ID users. Furthermore, all posts of multi-ID users are correlated, in particular, they do not occur too close together. Our algorithm detects the IDs whose posts display such statistical anomalies and identify them as coming from multi-ID users. statistical characteristics and identify them

Our experiments based upon the model of an open forum establish the feasibility of the statistical identification of multi-ID users. The accuracy of our algorithm depends on the length of time over which data is collected, and as ex-

pected, the more data is collected, the more accurate the results of the algorithm. Our error rates over long time periods are under 1%.

Related work. Very little work exists on determining the multi-ID users from open forum logs, such as chat rooms. However, a number of researchers have mined for various other information on chat rooms, instant messenging forums and internet relay forums, [1–7]

Paper outline. The remainder of this paper is as follows. First we introduce some preliminary definitions, followed by a model for generating forum logs (message postings). Section 3 contains a description of a multi-user identification algorithms and Section 4 presents the results of numerical simulations. We present the results with model-generated newsgroup as well as real newsgroup logs.

Acknowledgment We would like to thank Ahmet Camptepe who was responsible for collecting the chatroom logs that were used to illustrate some of the results in this paper. We would also like to thank Mukkai Krishnamoorthy, Sergej Roytman and Bülent Yener for useful conversations.

2 Preliminaries

In order to make the discussion more precise, we will introduce some definitions here. We use small letters $i, j, k, \ldots$ to denote specific IDs that post on the open forum, and capital letters $A, B, C, \ldots$ to denote specific actors. There is a (many to one) mapping $\mathcal{A}$ that associates IDs with actors, thus $\mathcal{A}(i) = A$ means that actor A is operating ID i. We use id to denote the inverse of $\mathcal{A}$, thus $\mathsf{id}(A) = \{i, j, k, \ldots\}$ is the set of IDs that that actor A is operating. The number of IDs that A is operating is given by $|\mathsf{id}(A)|$. If $|\mathsf{id}(A)| > 1$, then A is a multi-ID actor; otherwise, if $|\mathsf{id}(A)| = 1$, A is a single-ID actor.

We assume that there are relationships among the IDs, *i.e.*, an ID i can be the "friend" of one or more other IDs, $j_1, \ldots, j_k$. All these relationships are represented by the *friendship-graph* G, which is a graph in which all the IDs correspond to a vertex. There is an edge between two IDs if they are friends. We assume that, on the open forum, messages are only exchanged among friends. Two IDs that are not friends do not communicate during the time-period in question. We do not have access to the friendship graph, even though this graph governs the communication dynamics. We assume that every ID knows its friends.

A message posted by an ID has four attributes $\{t, \mathsf{id}_S, \mathsf{txt}, \mathsf{id}_R\}$. Here t is the time at which the message was posted; id_S is the source-ID that is posting the message; txt is the text of the message; and id_R is the set of IDs for the intended receivers of the message. For simplicity, assume that id_R always contains exactly one intended receiver. Our results, however, apply to the more general case. The time stamp is given by the server at the time it posts the message onto the screen.

We assume that the receiver-ID, upon seeing the message posted, knows that the message was intended for that ID. Not all four attributes are necessarily

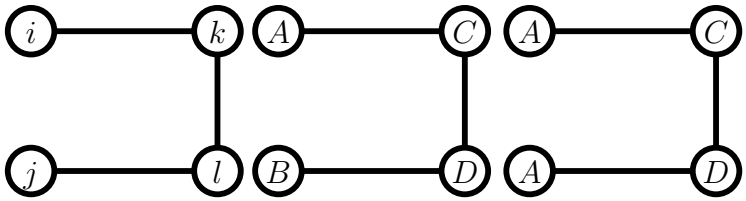
posted on the open forum. For example, in chatrooms, id_R is not posted and the receiver-ID knows implicitly if a message is meant for him/her from the text and context. In newsgroups, the id_R is often included in the post. We define the *forum log* $\mathcal{L}$ as the sequence of posts in the form $\{< t, \mathsf{id}_S >_i\}_{i=1}^{N}$, where N is the number of posts that were made. Note that we ignore the possible information that is present in the message texts, even though in some cases, the message reveals the receiver ID.

Our goal is to construct an algorithm to determine multi-ID actors using only the information in the forum log.

3 A Model of an Open Forum

We assume that the messages appear on a virtual screen in a sequence, and that they are accessible to all actors participating in the forum. In the reality though, these messages may appear on different physical screens. We do not address the motivation or the semantics of the messages, rather the stochastic process that generates the messages.

Underlying the communication is the friendship graph. For illustration, consider the friendship graph illustrated in the figure below.

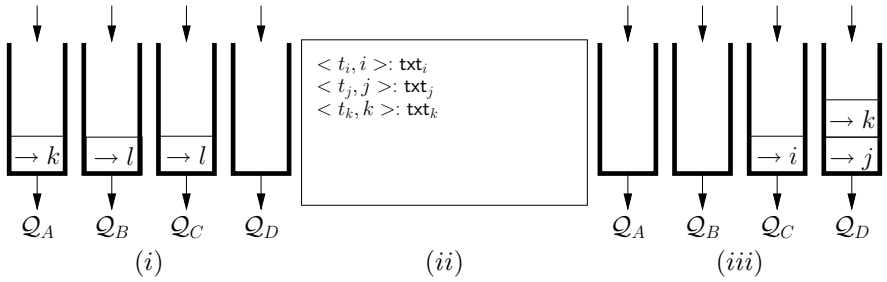


friendship graph (a)no multi-ID users (b)One multi-ID user

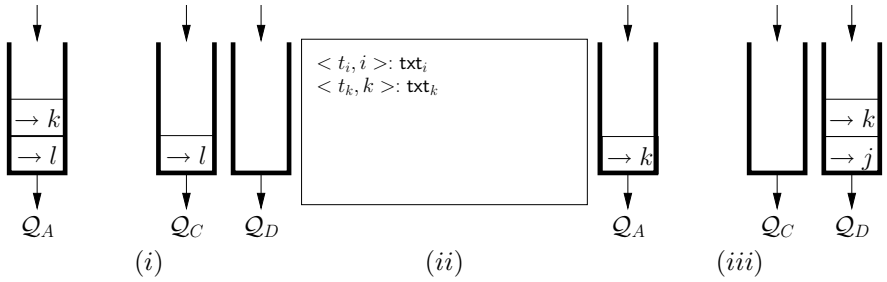
In (a), we show the graph after mapping each ID to its corresponding actor, thus for example, $\mathcal{A}(()j) = B$. In (b), we show a second scenario, in which now A is a multi-ID actor, thus $\mathsf{id}(A) = \{i, j\}$. Our main concern in this paper is to determine how the forum log will be different when the IDs are operated by actors as in (a) versus (b). The fundamental observation that will aid us toward this goal is that an actor has a finite bandwidth, i.e., it takes a user some amount of time to process messages onto the server. Thus, in (a), the messages for IDs i and j are processed by different actors, whereas in (b), a single actor has to divide her bandwidth between the two IDs. In particular, we will investigate the statistical consequences of this division of bandwidth among the IDs of a multi-ID actor.

To make this notion of bandwidth division more formal, we associate to each actor A, a processing queue, $\mathcal{Q}_A$. The messages that this actor wishes to post are placed in the queue, and are processed in FIFO order. Further, after completing a message, the actor submits it to the server, which is also implementing a queue. We assume that messages that arrive at the same time are places in any queues in a random order. To illustrate, notice that according to the friendship graph, three conversations are going on, namely $(i, k), (j, l), (k, l)$. Suppose that i, j and

k choose to initiate these conversations. When the actor graph is as in (a), the initial queue status is shown in the (i) in the figure below. (ii) shows the server messages after each actor has processed one message and (iii) shows the resulting queues after the actors see the messages and initiate replies into their queues.



The next figure shows the same evolution for the actor graph in (b). Notice how the forum log will be different solely on account of the fact that actor A is now operating more than one ID.



We now describe a forum log *generator* that implements such a model. Our generator can be made very general, but for illustration, we present one of the simplest versions. There are three components:

Initialization. Let $\{e_1, e_2, \ldots, e_m\}$ be a sequence of the edges in G (randomly ordered). For every actor A, a queue $\mathcal{Q}_A$ is defined; initially, all $\mathcal{Q}_A$s are empty. For every edge, one of its endpoints is randomly selected as *the source* s and the other endpoints is determined as the *recipient* r. Note that s and r are IDs, not actors. A reply message to r is pushed onto the queue $\mathcal{Q}_{\mathcal{A}(s)}$, i.e., onto the queue of the actor corresponding to the source ID. The result of the initialization step is an array of queues, $[\mathcal{Q}_{A_1}, \mathcal{Q}_{A_2}, \ldots, \mathcal{Q}_{A_n}]$, where each queue corresponds to unique actor operating on the forum. A queue may correspond to multiple IDs in the friendship graph G. Note that some queues may be empty.

Processing by Actors. Every actor A_i:

1. Processes and removes the first message on its queue. The time to process this message τ could be set randomly to simulate long and short messages. After processing this message, the actor submits it to the forum.

2. Scans the forum postings for any messages that are addressed to any ID in $\mathsf{id}(A_i)$. Each such message generates a reply to the poster of the message. This reply is pushed onto $\mathcal{Q}_{A_i}$.

Processing by Forum. The forum has a global queue $\mathcal{Q}_F$ of messages to be posted. The messages arrive according to the times they were submitted to the forum by actors; if a posts arrive at the same time, the forum sorts them into an arbitrary (random) order. The forum processes its queue using FIFO order, taking a time of 1 unit to post a message.

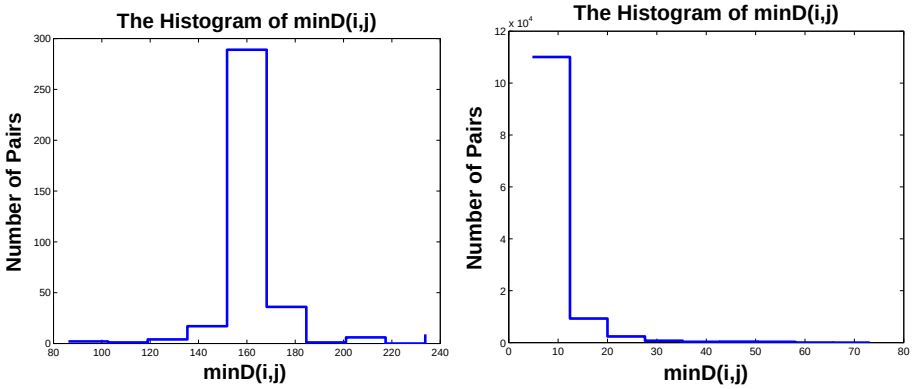
4 Algorithms.

The input to the multiuser-identification algorithm is a the forum log $\mathcal{L} = \{< t, \mathsf{id}_S >_i\}_{i=1}^N$, the times of the postings and the IDs that made the postings. To design an identification algorithm, we need to determine a statistical property of the communication exchange which separates multi-users from users that employ one ID only. The intuition behind out algorithm is that since a user has only one queue, it can only process messages sequentially. This is independent of whether she is operating one ID or multiple IDs. Suppose that, on average, it takes an actor τ_0 to complete a message. Then, the time gap between two messages posted under different IDs but by the same actor will on average be τ_0 time units. On the other hand, if two different actors are posting messages for a pair of IDs, then this restriction does not hold. In fact, over a long enough time period, one expects that the posts of these two IDs may arrive arbitrarily close to each other.

Let i, j be two IDs, and consider the two subsequences of the forum log consisting only of the posts of each of these IDs: $\{< t_i, \mathsf{i} >\}$ and $\{< t_j, \mathsf{j} >\}$. Define the set of separation times, $\{D_{ij}\}$, as the set of time differences between *consecutive* posts of the two IDs – i.e., for every pair of times t_i, t_j at which i posts followed by j or j followed by i, and there are no posts made in between these two posts, then $|t_j - t_i| \in \{D_{ij}\}$. We define two separation indices, the *mean separation index* $M(i, j)$ for IDs i, j, and the *minimum separation index* $\min D(i, j)$:

$$M(i, j) = \text{mean}\{D_{ij}\}$$
$$\min D(i, j) = \min\{D_{ij}\}$$

We expect that if $\mathcal{A}(i) = \mathcal{A}(j)$, then these separation indices will be significantly larger that if $\mathcal{A}(i) \neq \mathcal{A}(j)$. More specifically, if $M(i, j)$ and $\min D(i, j)$ are small, then it is not possible that $\mathcal{A}(i) = \mathcal{A}(j)$. On the other hand, if they are large, then it would be extremely unlikely that $\mathcal{A}(i) \neq \mathcal{A}(j)$ on account of the independence of the actors behind the IDs, and hence it is likely that $\mathcal{A}(i) = \mathcal{A}(j)$. The intuition we have described would hold for any model of the forum log that assumes a finite bandwidth for the actors, as well as sequential processing of messages. The quantities that would vary from model to model would be exactly how large the separation indices would have to get before one could declare that a pair of IDs is suspicious and is probably from a multi-ID actor. In order to test these



(a) (b)

Fig. 1. Histograms of $\min D(i,j)$. In (a), $\mathcal{A}(i) = \mathcal{A}(j)$; in (b), $\mathcal{A}(i) \neq \mathcal{A}(j)$

hypotheses, we simulate a forum log according to the model in the previous section and compute the statistics $M(i,j)$ and $\min D(i,j)$ for every pair of IDs. We give, in the Table 1 below, the averages of these statistics over pairs of IDs from the same actor versus pairs from different actors separately. The statistics

	$\mathcal{A}(i) = \mathcal{A}(j)$	$\mathcal{A}(i) \neq \mathcal{A}(j)$
$\underset{i,j}{\text{mean}}\{M(i,j)\}$	382.79	264.63
$\underset{i,j}{\text{mean}}\{\min D(i,j)\}$	156.36	3.22

Table 1. Some statistical properties of the $\{D_{ij}\}$

in Table 1 were obtained assuming that the time τ that an actor takes to prepare a post is 250 units. It is clear that the separation indices are drastically different depending on whether the pair of actors is on the same versus on different actors. Further, the histograms in Figure 1 indicate that not only are the separation indices different on average, but the distributions are also well separated. We are thus led to the following algorithm for identifying multi-ID actors:

1: //**Algorithm to identify multi-ID actors**
2: //**Input:** Forum Log $\mathcal{L} = \{< t, \mathsf{id}_S >_i\}_{i=1}^N$.
3: //**Output:** Pairs of IDs on the same actor.
4: For every pair of IDs i, j, compute $\min D(i,j)$;
5: Cluster the values of $\min D(i,j)$ into two groups, G_{large} containing the large values, and G_{small} containing the smaller. Every pair (i,j) belongs to one of these groups.
6: **return** all the pairs $(i,j) \in G_{large}$;

Defining the equivalence relation $i \equiv j$ iff $(i, j) \in G_{large}$, the equivalence classes of the IDs then partitions the IDs into sets, each of which are operated by one actor. Thus we can identify all the IDs common to a given actor from the algorithm above.

5 Experiments.

The model for the dynamics of the forum log has the following parameters

del: The average time for an actor to compose a message and submit it to the forum server. Although, in general, this average composition time can be time and/or actor dependent, for our simplified model, we assumed it to be a constant.

wid: A parameter specifying how much the actual time to compose a message can vary from the average time *del*. If τ is the time to compose a message, then we assume that τ is uniformly distributed in $\left[del - \frac{1}{2}wid, del + \frac{1}{2}wid\right]$. In general, *wid* can be time and/or actor dependent, but for our simplified model, we keep *wid* fixed. *wid* can be viewed as a noise parameter that introduces some non-determinism into the forum log.

len: the length of a post by a actor; it is assumed that *len* determines the minimal time-period needed for a actor to compose the message; in general, *len* is time- and actor-dependent, but for this model, *len* is a constant.

run: the total number of time units for which the forum log is generated.

nID: the total number of IDs participating in the communication exchange.

maxID: the maximum number of IDs employed by a single actor – a *k-ID actor* operates k IDs.

nFriend: the average number of friends an ID has in the friendship graph.

In our simulations, we fixed $nID = 500$, with about an equal number of 1-ID actors, 2-ID actors, 3-ID actors and 4-ID actors, thus $maxID = 4$. We fixed $nFriend = 5$, and used different values of *del*, *wid* and *run* to determine how these three parameters influence the accuracy of the detection algorithm, which we define as the percentage of pairs of IDs that are assigned to the correct group (multi or single).

$$Accuracy = \frac{\text{The number of pairs (i,j) assigned into correct group}}{\text{The total number of (i,j) pairs}}$$

To implement step 5 of the algorithm, we used a standard K-means algorithm [8], with K set to 2.

The details of the simulation are as follows. First randomly generate a friendship graph with average degree 5. Using this friendship graph, we run the forum generator for *run* timesteps, to generate a forum log. We then implement the detection algorithm to determine which ID's are from single-ID actors and which IDs are on the same actor. We then compute the accuracy, and repeat this entire simulation over 10 times to get a more accurate estimate of the average accuracy.

Table 2 illustrates how the accuracy depends on *wid* and *del* when $run = 1,000,000$ and when $run = 10,000$. When the observation sequence is long enough ($run = 1,000,000$), the accuracy is almost 100% and is not influenced much by *wid*, i.e, the fact that messages take random amounts of time to compose does not seem to heavily affect the algorithm's accuracy. However, there is a slight decrease in performance when *del* increases. This is mostly due to the fact that there are fewer posts (data) when *del* increases, as the observation period is fixed. Showed in *Figure* 2 is the dependence of the accuracy on *del*, the time to compose a message, for different valuse of the noise parameter *wid*.

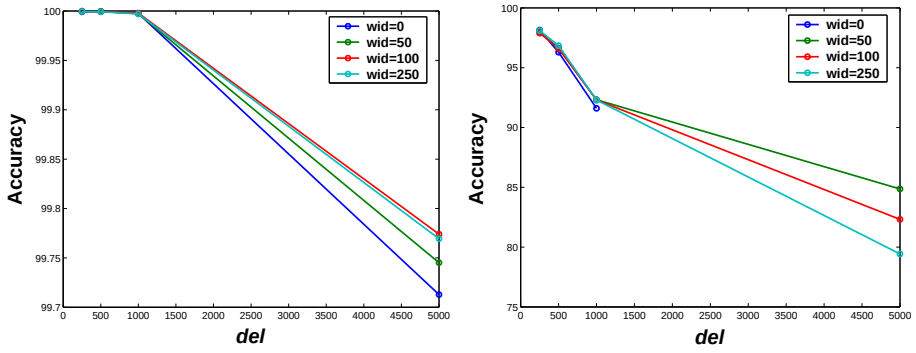
	wid			
del	0	50	100	250
250	99.9992	99.9998	99.9999	99.9996
500	99.9995	99.9997	99.9993	99.9993
1000	99.9976	99.9972	99.9979	99.9972
5000	99.7129	99.7452	99.7740	99.7695

(a) $run = 1,000,000$

	wid			
del	0	50	100	250
250	98.1425	97.98	97.8906	98.0964
500	96.2923	96.634	96.6173	96.8459
1000	91.6025	92.3209	92.3017	92.2991
5000	—	84.8614	82.3213	79.4296

(b)$run = 10,000$

Table 2. The dependence of the Accuracy (in %) on del, wid, run.



(a) $run = 1,000,000$ (b) $run = 10,000$

Fig. 2. The dependence of the Accuracy (in %) on *del* and *wid*

Notice that when *run* is small, i.e., the observation period is small, the accuracy considerably drops. This is due to the significant drop in the amount of data available for the classification into single versus multi-ID actors. This is illustrated in Table 5, where we show the number of posts that the various types of actors make. Also

illustrated in Table 5 is the effect of the limited bandwidth assumption that we place on the actors. The 4-ID actors post less frequently than the 1-ID actors. While one would expect the frequency of posting to have dropped by a factor of 4 in going from the 1-ID actors to the 4-ID actors, it is not quite a factor of 4. The reason is that some of the 1-ID actors are friends with IDs belonging to 4-ID actors, which means that the frequency of posting of these 1-ID actors will be slowed down by the fact that they have to wait longer for the responses from the 4-ID actors.

	# of IDs operated by an actor			
del	1	2	3	4
250	2876.6	1910.8	1326.4	1015.9
500	1436.6	954.8	663.3	507.3
1000	717.1	477.2	331.5	253.8
5000	142.9	95.7	66.6	50.7

(a) $run = 1,000,000$

	# of IDs operated by an actor			
del	1	2	3	4
250	23.95	18.89	13	9.75
500	12.41	9.18	6.31	4.75
1000	5.39	4.23	2.98	2.25
5000	0.38	0.49	0.33	0.25

(b) $run = 10,000$

Table 3. The average number of messages posted by different types of actors.

In principle the average number of posts could be used to discriminate between multi-ID actors and single-ID actors, however the distributions of this statistic are not as separated as with min$D$.

6 Conclusions.

We have presented an algorithm to identify multi-ID users, the justification of which is based on a novel and reasonable model of communication exchange on a public forum. Along this direction, the model could be considerably expanded to include more realistic and nondeterministic phenomena, such as multi-party exchange; server delays; and different composition time distributions for each actor. Further, one could allow the friendship graphs and the general communication dynamics on the forum to be time varying. It is also possible to incorporate statistical parameters of the real communication forum into the model.

What we have demonstrated is that under the broad assumption of a finite processing power for each actor, and the fact that messages are processed sequentially, an actor operating multiple IDs will give itself away by the fact that those IDs will have different posting statistics to the normal, single-ID actors. In particular the posts of these IDs will not be independent, nor as frequent. We see that introducing randomness into the time to compose a message does not have much effect on the algorithm, nor did changing the average time to compose a message – *i.e.*, the algorithm is quite robust to the specific details of the model, which is comforting. Our simulations show that if the time to compose a message is 5000 units (a unit being the time for the server to process a message), then with 1,000,000 time units of observation, we can essentially obtain 100% accuracy. To put this in perspective, in a chatroom if it takes about 10 seconds to compose a message, then we need under 1 hour of observation.

It is, of course, important to realize that every identification algorithm can potentially be deceived by a skillful actor who intends to hide its multiple IDs. However the

attempt to deceit our algorithm will likely cause a time delay in the postings of the user (or other irregularities). A systematic delay in a user's activity may in turn be employed for the identification purposes.

References

1. Nardi, B.A., Whittaker, S., Bradner, E.: Interaction and outeraction: instant messaging in action. In: Computer Supported Cooperative Work. (2000) 79–88
2. We, T., Khan, F.M., Fisher, T.A., Shuler, L.A., Pottenger, W.M.: Error-driven boolean-logic-rule-based learning for mining chat-room conversations (2002)
3. Khan, F.M., Fisher, T.A., Shuler, L., Wu, T., Pottenger, W.M.: Mining chat-room conversations for social and semantic interactions (2002)
4. Budzik, J., Bradshaw, S., Fu, X., Hammond, K.: Clustering for opportunistic communication. In: Proceedings of WWW 2002, ACM Press (2002)
5. Elizabeth, R.: Electropolis: Communication and community on internet relay chat (1992)
6. Whittaker, S., Jones, Q., Terveen, L.: Contact management: Identifying contacts to support long-term communication (2002)
7. Isaacs, E., Kamm, C., Schiano, D.J., Walendowski, A., Whittaker, S.: Characterizing instant messaging from recorded logs (2002)
8. Bishop, C.M.: Neural Networks for Pattern Recognition. Clarendon Press, Oxford (1995)