Humans and Bots in Internet Chat: Measurement, Analysis, and Automated Classification

Steven Gianvecchio, Mengjun Xie, Zhenyu Wu, and Haining Wang

***Abstract*—The abuse of chat services by automated programs, known as chat bots, poses a serious threat to Internet users. Chat bots target popular chat networks to distribute spam and malware. In this paper, we first conduct a series of measurements on a large commercial chat network. Our measurements capture a total of 16 different types of chat bots ranging from simple to advanced. Moreover, we observe that human behavior is more complex than bot behavior. Based on the measurement study, we propose a classification system to accurately distinguish chat bots from human users. The proposed classification system consists of two components: (1) an entropy-based classifier and (2) a Bayesian-based classifier. The two classifiers complement each other in chat bot detection. The entropy-based classifier is more accurate to detect unknown chat bots, whereas the Bayesian-based classifier is faster to detect known chat bots. Our experimental evaluation shows that the proposed classification system is highly effective in differentiating bots from humans.**

***Index Terms*—Bots, Internet chat, measurement, classification.**

I. Introduction

Internet chat is a popular application that enables real-time text-based communication. Millions of people around the world use Internet chat to exchange messages and discuss a broad range of topics online. Internet chat is also a unique networked application, because of its human-to-human interaction and low bandwidth consumption [1]. However, the large user base and open nature of Internet chat make it an ideal target for malicious exploitation.

The abuse of chat services by automated programs, known as *chat bots*, poses a serious threat to online users. Chat bots have been found on a number of chat systems, including large commercial chat networks, such as AOL [2], Yahoo! [3]–[5] and MSN [6], and open chat networks, such as IRC and Jabber. There are also reports of bots in some non-chat systems with chat features, including online games, such as World of Warcraft [7]. Chat bots exploit these online systems to send spam, spread malware, and mount phishing attacks.

So far, the efforts to combat chat bots have focused on two different approaches: (1) keyword-based filtering and (2) human interactive proofs. The keyword-based message filters, used by third party chat clients [8], suffer from high false negative rates because bot makers frequently update chat bots to evade published keyword lists. The use of human interactive proofs, such as CAPTCHAs [9], is also ineffective because bot operators assist chat bots in passing the tests to log into chat rooms [4], [5]. In August 2007, Yahoo! implemented CAPTCHA to block bots from entering chat rooms, but bots are still able to enter chat rooms in large numbers. There are online petitions against both AOL and Yahoo! [2], requesting that the chat service providers address the growing bot problem. While online systems are besieged with chat bots, no systematic investigation on chat bots has been conducted. The effective detection system against chat bots[1] is in great demand but still missing.

In the paper, we first perform a series of measurements on a large commercial chat network, Yahoo! chat, to study the behaviors of chat bots and humans in online chat systems. Our measurements capture a total of 16 different types of chat bots. The different types of chat bots use different triggering mechanisms and text obfuscation techniques. The former determines message timing, and the latter determines message content. Our measurements also reveal that human behavior is more complex than bot behavior, which motivates the use of entropy rate, a measure of complexity, for chat bot classification. Based on the measurement study, we propose a classification system to accurately distinguish chat bots from humans. There are two main components in our classification system: (1) an entropy classifier and (2) a Bayesian classifier. Based on the characteristics of message time and size, the entropy classifier measures the complexity of chat flows and then classifies them as bots or humans. In contrast, the Bayesian classifier is mainly based on message content for detection. The two classifiers complement each other in chat bot detection. While the entropy classifier requires more messages for detection and, thus, is slower, it is more accurate to detect unknown chat bots. Moreover, the entropy classifier helps train the Bayesian classifier. The machine learning classifier requires less messages for detection and, thus, is faster, but cannot detect most unknown bots. By combining the entropy classifier and the Bayesian classifier, the proposed classification system is highly effective to capture chat bots, in terms of accuracy and speed. We conduct experimental tests on the classification system, and the results validate its efficacy on chat bot detection.

The remainder of this paper is structured as follows. Section II covers background on chat bots and related work. Section

S. Gianvecchio (gianvecchio@mitre.org) is with the MITRE Corporation, McLean, VA, 22102. The author's affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author.

M. Xie (mxxie@ualr.edu) is with the Department of Computer Science, University of Arkansas at Little Rock, Little Rock, AR 72204.

Z. Wu (adamwu@cs.wm.edu) and H. Wang (hnw@cs.wm.edu) are with the Department of Computer Science, College of William and Mary, Williamsburg, VA, 23185.

[1] Those bots in botnets that exploit chat platforms, such as IRC, as a command and control mechanism are different from the chat bots interacting with real humans, and are not the focus of this paper.

III details our measurements of chat bots and humans. Section IV describes our chat bot classification system. Section IV-B evaluates the effectiveness of our approach to chat bot detection. Finally, Section V concludes the paper and discusses directions for our future work.

II. Background and Related Work

A. Chat Systems

Internet chat is a real-time communication tool that allows on-line users to communicate via text in virtual spaces, called chat rooms or channels. There are a number of protocols that support chat [10], including IRC, Jabber/XMPP, MSN/WLM (Microsoft), OSCAR (AOL), and YCHT/YMSG (Yahoo!). The users connect to a chat server via chat clients that support a certain chat protocol, and they may browse and join many chat rooms featuring a variety of topics. The chat server relays chat messages to and from on-line users. A chat service with a large user base might employ multiple chat servers. In addition, there are several multi-protocol chat clients, such as Pidgin (formerly GAIM) and Trillian, that allow a user to join different chat systems.

Although IRC has existed for a long time, it has not gained mainstream popularity. This is mainly because its console-like interface and command-line-based operation are not user-friendly. The recent chat systems improve user experience by using graphic-based interfaces, as well as adding attractive features such as avatars, emoticons, and audio-video communication capabilities. Our study is carried out on the Yahoo! chat network, one of the largest and most popular commercial chat systems.

Yahoo! chat uses proprietary protocols, in which the chat messages are transmitted in plain-text, while commands, status and other meta data are transmitted as encoded binary data. Unlike those on most IRC networks, users on the Yahoo! chat network cannot create chat rooms with customized topics because this feature is disabled by Yahoo! to prevent abuses [11]. In addition, users on Yahoo! chat are required to pass a CAPTCHA word verification test in order to join a chat room. This recently-added feature is to guard against a major source of abuse—bots.

B. Chat Bots

The term *bot*, short for robot, refers to automated programs, that is, programs that do not require a human operator. A chat bot is a program that interacts with a chat service to automate tasks for a human, e.g., creating chat logs. The first-generation chat bots were designed to help operate chat rooms, or to entertain chat users, e.g., quiz or quote bots. However, with the commercialization of the Internet, the main enterprise of chat bots is now sending chat spam. Chat bots deliver spam URLs via either links in chat messages or user profile links. A single bot operator, controlling a few hundred chat bots, can distribute spam links to thousands of users in different chat rooms, making chat bots very profitable to the bot operator who is paid per-click through affiliate programs. Other potential abuses of chat bots include spreading malware, phishing, booting,[2] and other malicious activities.

A few countermeasures have been used to defend against the abuse of chat bots, though none of them are very effective. On the server side, CAPTCHA tests are used by Yahoo! chat in an effort to prevent chat bots joining chat rooms. However, this defense becomes ineffective as chat bots bypass CAPTCHA tests with human assistance. We have observed that bots continue to join chat rooms and sometimes even become the majority members of a chat room after the deployment of CAPTCHA tests. Third-party chat clients filter out chat bots, mainly based on key words or key phrases that are known to be used by chat bots. The drawback with this approach is that it cannot capture those unknown or evasive chat bots that do not use the known key words or phrases.

C. Related Work

Dewes et al. [1] conducted a systematic measurement study of IRC and web-chat traffic, revealing several statistical properties of chat traffic. (1) Chat sessions tend to last for a long time, and a significant number of IRC sessions last much longer than web-chat sessions. (2) Chat session inter-arrival time follows an exponential distribution, while the distribution of message inter-arrival time is not exponential. (3) In terms of message size, all chat sessions are dominated by a large number of small packets. (4) Over an entire session, typically a user receives about 10 times as much data as he sends. However, very active users in web-chat and automated scripts used in IRC may send more data than they receive.

There is considerable overlap between chat and instant messaging (IM) systems, in terms of protocol and user base. In general, chat refers to a system that supports chat rooms or channels, e.g., IRC, whereas IM refers to a system that supports direct messaging and presence, e.g., AIM. Many widely used chat systems such as IRC predate the rise of IM systems, and have great impact upon the IM system and protocol design. In return, some new features that make the IM systems more user-friendly have been back-ported to the chat systems. For example, IRC, a classic chat system, implements a number of IM-like features, such as presence and file transfers, in its current versions. Some messaging service providers, such as Yahoo!, offer both chat and IM accesses to their end-user clients. With this in mind, we outline some related work on IM systems. Liu et al. [12] explored client-side and server-side methods for detecting and filtering IM spam or *spim* for short. However, their evaluation is based on a corpus of short email spam messages, due to the lack of data on spim. In [13], Mannan et al. studied IM worms, automated malware that spreads on IM systems using the IM contact list. Leveraging the spreading characteristics of IM malware, Xie et al. [14] presented an IM malware detection and suppression system based on the honeypot concept. Similarly, Trivedi et al. [15] used honeypots to analyze network and content characteristics of spim. Although not directly related to chat or instant messaging, Jonathan et. al [16] discuss the problem of socially-interactive malware.

[2] The term booting is chat-speak for causing a user to disconnect from chat.

Botnets consist of a large number of slave computing assets, which are also called “bots”. However, the usage and behavior of bots in botnets are quite different from those of chat bots. The bots in botnets are malicious programs designed specifically to run on compromised hosts on the Internet, and they are used as platforms to launch a variety of illicit and criminal activities such as credential theft, phishing, distributed denial-of-service attacks, and other attacks. In contrast, chat bots are automated programs designed mainly to interact with chat users by sending spam messages and URLs in chat rooms. Although having been used by botnets as command and control mechanisms [17], IRC and other chat systems do not play an irreplaceable role in botnets. In fact, due to considerable effort and progress on detecting and thwarting IRC-based botnets [18]–[20], the control architectures of more recent botnets, such as Zeus, Koobface, and Conficker (or Storm), are P2P or HTTP-based [21]–[23], instead of IRC-based.

Chat spam shares some similarities with email spam. Like email spam, chat spam contains advertisements of illegal services and counterfeit goods, and solicits human users to click spam URLs. Chat bots employ many text obfuscation techniques used by email spam such as word padding and synonym substitution. Since the detection of email spam can be easily converted into the problem of text classification, many content-based filters utilize machine-learning algorithms for filtering email spam. Among them, Bayesian-based statistical approaches [24]–[27] have achieved high accuracy and performance. Although very successful, Bayesian-based spam detection techniques still can be evaded by carefully crafted messages [28]–[30].

III. Measurement

In this section, we detail our measurements on Yahoo! chat, one of the most popular commercial chat services. The focus of our measurements is on public messages posted to Yahoo! chat rooms. The logging of chat messages is available on the standard Yahoo! chat client, as well as most third-party chat clients. Upon entering chat, all chat users are shown a disclaimer from Yahoo! that other users can log their messages. However, we consider the contents of the chat logs to be sensitive, so we only present fully-anonymized statistics.

Our data was mainly collected between August and November of 2007. In late August of 2007, Yahoo! implemented a CAPTCHA check on entering chat rooms [4], [31], creating technical problems that made their chat rooms unstable for about two weeks [32], [33]. At the same time, Yahoo! implemented a protocol update, preventing most third-party chat clients, used by a large proportion of Yahoo! chat users, from accessing the chat rooms. In short, these upgrades made the chat rooms difficult to be accessed for both chat bots and humans. In mid to late September of 2007, both chat bot and third-party client developers updated their programs. By early October of 2007, chat bots were again found in Yahoo! chat [5], possibly bypassing the CAPTCHA check with human assistance. Due to these problems and the lack of chat bots in September and early October of 2007, we perform our main analysis on August 2007 and November 2007 chat logs.

In August and November of 2007, we collected a total of 1,440 hours of chat logs by passively monitoring different chat rooms. The data collected includes 147 separate chat logs from 21 different chat rooms. The chat rooms were selected at random from the most popular topic areas, including culture, health, music, politics, religion, and romance. Since Yahoo! enforces a session limit of three hours and specific rooms are often full, we sometimes had to join a different room for the same topic after being disconnected, e.g., “Politics Lobby 4” instead of “Politics Lobby 3.” The chat logs are supplemented by 64 hours of additional chat logs from October of 2008 on advanced responder bots. The process of reading and labeling the chat logs required about 100 hours per month of data. To the best of our knowledge, we are the first in the large scale measurement and classification of chat bots.

A. Log-Based Classification

In order to characterize the behavior of human users and that of chat bots, we need two sets of chat logs pre-labeled as bots and humans. To create such datasets, we perform log-based classification by reading and labeling a large number of chat logs. The chat users are labeled in three categories: human, bot, and ambiguous. The labeled datasets are used as the ground truth of this work.

The log-based classification process is a variation of the Turing test. In a standard Turing test [34], the examiner converses with a test subject (a possible machine) for five minutes, and then decides if the subject is a human or a machine. In our classification process, the examiner observes a long conversation between a test subject (a possible chat bot) and one or more third parties, and then decides if the subject is a human or a chat bot. In addition, our examiner checks the content of URLs and typically observes multiple instances of the same chat bot, which further improve our classification accuracy. Moreover, given that the best practices of current artificial intelligence (AI) [35] can rarely pass a non-restricted Turing test, i.e., a Turing test that is not limited to a specific subject, our classification of chat bots by a human expert should be very accurate. Meanwhile, we readily acknowledge that any possible biases or errors in the ground truth data are likely to have negative effects on our analysis and evaluation results.

Although a Turing test is subjective, we outline a few important criteria. The main criterion for being labeled as human is a high proportion of specific, intelligent, and human-like responses to other users. In general, when a user’s responses would require more advanced intelligence than current state-of-the-art AI [35], then the user is labeled as human. The ambiguous label is reserved for non-English, incoherent, or non-communicative users. The criteria for being classified as bot are as follows. The first is the lack of the intelligent responses required for the human label. The second is the repetition of similar phrases either over time or from other users (other instances of the same chat bot). The third is the presence of spam or malware URLs in messages or in the user’s profile.

B. Statistical Analysis

In total, our measurements capture 16 different types of chat bots. The type of a chat bot is determined by its triggering mechanisms and text obfuscation schemes. The former relates to message timing, and the latter relates to message content. The two main types of triggering mechanisms observed in our measurements are timer-based and response-based. A timer-based bot sends messages based on a timer, which can be periodic (i.e., fixed time intervals) or random (i.e., variable time intervals). A response-based bot sends messages based on programmed responses to specific content in messages posted by other users.

There are many different kinds of text obfuscation schemes. The purpose of text obfuscation is to vary the content of messages and make bots more difficult to recognize or appear more human-like. We observe four basic text obfuscation methods that chat bots use to evade filtering or detection. First, chat bots introduce random characters or space into their messages, similar to some spam emails. Second, chat bots use various synonym phrases to avoid listed keywords. By this method, a template with several synonyms for multiple words can lead to thousands of possible messages. Third, chat bots use short messages or break up long messages into multiple messages to evade message filters that work on a message-by-message basis. Fourth, and most interestingly, chat bots replay human phrases entered by other chat users.

According to our observation, the main activity of chat bots is to send spam links to chat users. There are two approaches that chat bots use to distribute spam links in chat rooms. The first is to post a message with a spam link directly in the chat room. The second is to enter a spam URL in the chat bot's user profile and then convince users to view the profile and click the link. Our logs also include some examples of malware spreading via chat rooms. The behavior of malware-spreading chat bots is very similar to that of spam-sending chat bots, as both attempt to lure human users to click links. Although we did not perform detailed malware analysis on links posted in the chat rooms and Yahoo! applies filters to block links to known malicious files, we found several worm instances in our data. There are 12 W32.Imaut.AS [36] worms appeared in the August 2007 chat logs, and 23 W32.Imaut.AS worms appeared in the November 2007 chat logs. The November 2007 worms' attempts to send links to malicious code were filtered by Yahoo!, the messages appeared with the links removed. However, the August 2007 worms were able to send out malicious links. Interestingly, there were no unambigously benign chat bots in the data. In other words, all of the chat bots in the data were involved in at least one of the following activities: posting spam links in the channel, repeatedly referring users to their profile, spamming political or religious messages, or attempting to spread malware.

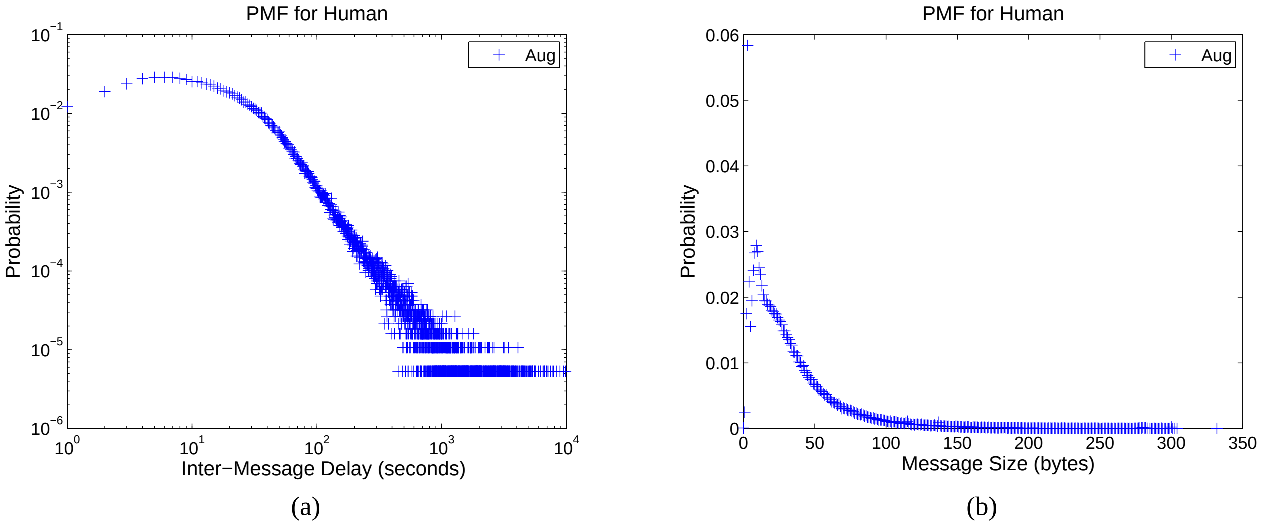
The focus of our measurements is mainly on short term statistics, as these statistics are most likely to be useful in chat bot classification. The two key measurement metrics in this study are inter-message delay and message size. Based on these two metrics, we profile the behavior of humans and that of chat bots. Among chat bots, we further divide them into six different groups: periodic bots, random bots, responder bots, replay bots, replay-responder bots, and advanced responder bots. With respect to these short-term statistics, humans and chat bots behave differently, as shown below.

1) Humans: Figure 1 shows the probability distributions of human inter-message delay and message size. Since the behavior of humans is persistent, we only draw the probability mass function (PMF) curves based on the August 2007 data. The previous study on Internet chat systems [1] observed that the distribution of inter-message delay in chat systems was heavy tailed. In general our measurement result conforms to that observation. The body part of the PMF curve in Figure 1 (a) (log-log scale) can be linearly fitted, indicating that the distribution of human inter-message delays follows a power law. In other words, the distribution is heavy tailed. We also find that the PMF curve of human message size in Figure 1 (b) can be well fitted by an exponential distribution with $\lambda = 0.034$ after excluding the initial spike.

2) Periodic Bots: A periodic bot posts messages mainly at regular time intervals. The delay periods of periodic bots, especially those bots that use long delays, may vary by several seconds. The variation of delay period may be attributed to either transmission delay caused by network traffic congestion or chat server delay, or message emission delay incurred by system overloading on the bot hosting machine. The posting of periodic messages is a simple but effective mechanism for distributing messages, so it is not surprising that a substantial portion of chat bots use periodic timers.

We display the probability distributions of inter-message delay and message size for periodic bots in Figure 2. We use '+' for displaying August 2007 data and '•' for November 2007 data. The distributions of periodic bots are distinct from those of humans shown in Figure 1. The distribution of inter-message delay for periodic bots clearly manifests the timer-triggering characteristic of periodic bots. There are three clusters with high probabilities at time ranges [30-50], [100-110], and [150-170]. These clusters correspond to the November 2007 periodic bots with timer values around 40 seconds and the August periodic bots with timer values around 105 and 160 seconds, respectively. The message size PMF curve of the August periodic bots shows an interesting bell shape, much like a normal distribution. After examining message contents, we find that the bell shape may be attributed to the message composition method some August bots used. As shown in Appendix A, some August periodic bots compose a message using a single template. The template has several parts and each part is associated with several synonym phrases. Since the length of each part is independent and identically distributed, the length of whole message, i.e., the sum of all parts, should approximate a normal distribution. The November 2007 bots employ a similar composition method, but use several templates of different lengths. Thus, the message size distribution of the November 2007 periodic bots reflects the distribution of the lengths of the different templates, with the length of each individual template approximating a normal distribution.

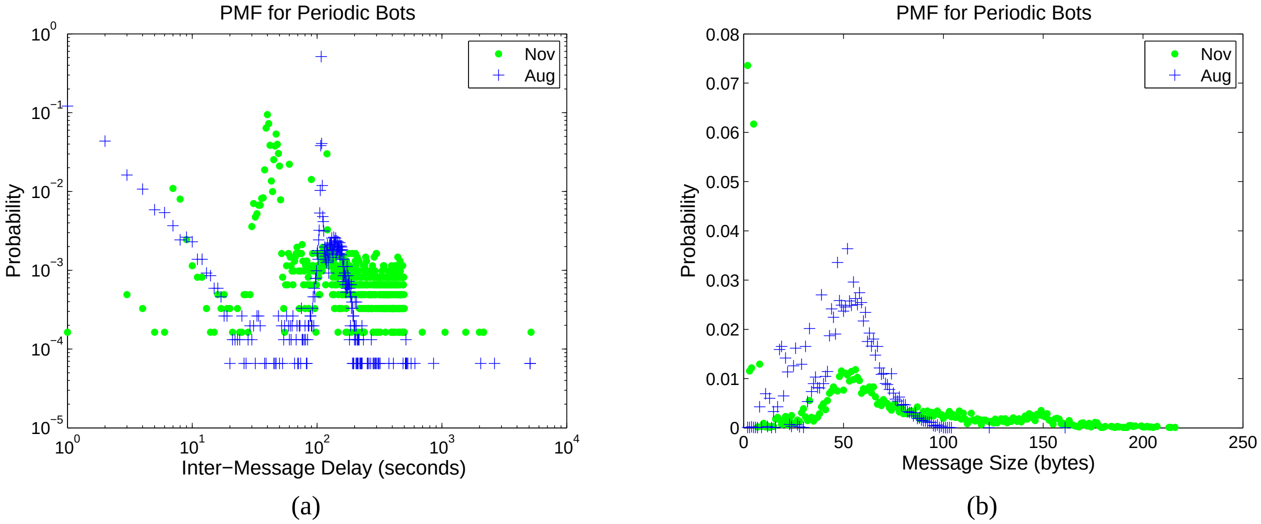
3) Random Bots: A random bot posts messages at random time intervals. The random bots in our data used different



(a)

(b)

Fig. 1. Distribution of human inter-message delay (a) and message size (b)



(a)

(b)

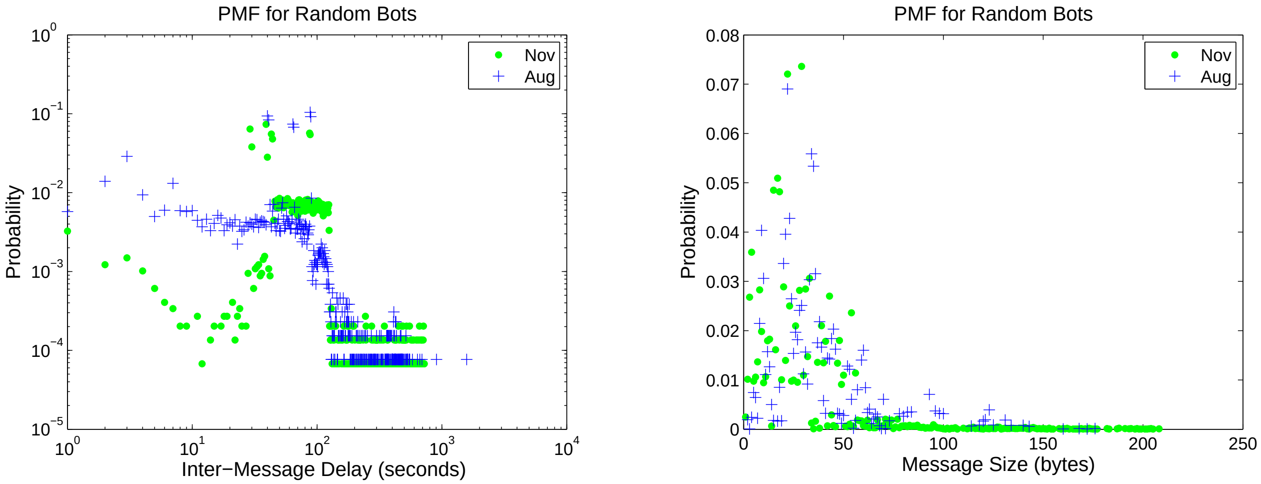
Fig. 2. Distribution of periodic bot inter-message delay (a) and message size (b)

random distributions, some discrete and others continuous, to generate inter-message delays. The use of random timers makes random bots appear more human-like than periodic bots. In statistical terms, however, random bots exhibit quite different inter-message delay distributions than humans.

Figure 3 depicts the probability distributions of inter-message delay and message size for random bots. Compared to periodic bots, random bots have more dispersed timer values. In addition, the August 2007 random bots have a large overlap with the November 2007 random bots. The points with high probabilities (greater than 10^{-2}) in the time range [30-90] in Figure 3 (a) represent the August 2007 and November 2007 random bots that use a discrete distribution of 40, 64, and 88 seconds. The wide November 2007 cluster with medium probabilities in the time range [40-130] is created by the November 2007 random bots that use a uniform distribution between 45 and 125 seconds. The probabilities of different message sizes for the August 2007 and November 2007 random bots are mainly in the size range [0-50]. Unlike periodic bots, most random bots do not use template or synonym replacement, but directly repeat messages. Thus, as their messages are selected from a database at random, the message size distribution reflects the proportion of messages of different sizes in the database.

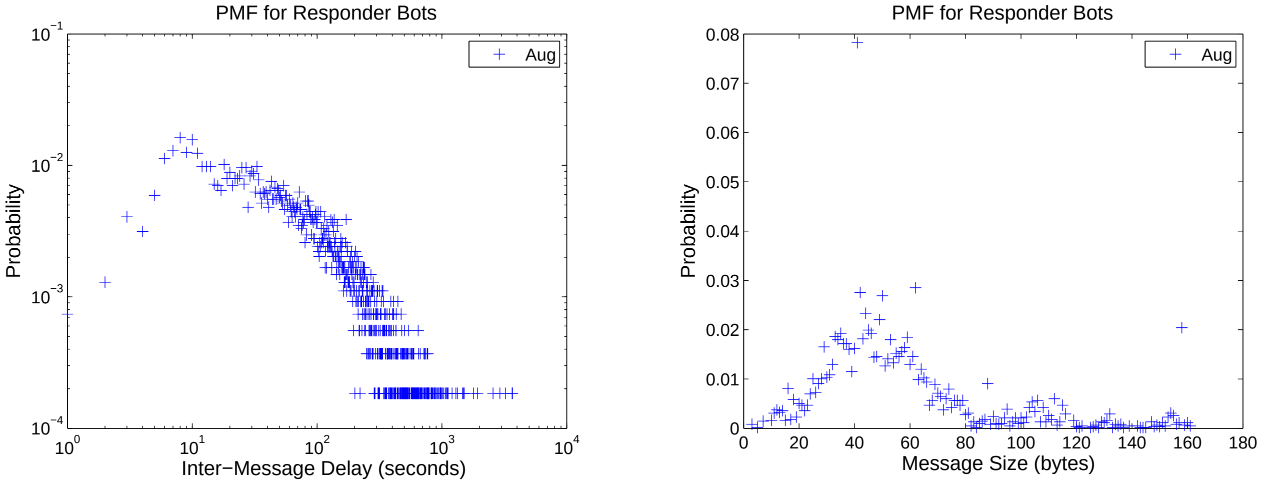
4) Responder Bots: A responder bot sends messages based on the content of messages in the chat room. For example, a message ending with a question mark may trigger a responder bot to send a vague response with a URL, as shown in Appendix A. The vague response, in the context, may trick human users into believing that the responder is a human and further clicking the link. Moreover, the message triggering mechanism makes responder bots look more like humans in terms of timing statistics than periodic or random bots.

To gain more insights into responder bots, we managed to obtain a configuration file for a typical responder bot [37]. There are a number of parameters for making the responder bot mimic humans. The bot can be configured with a fixed typing rate, so that responses with different lengths take different time to “type.” The bot can also be set to either ignore triggers while simulating typing, or rate-limit responses. In



(a) (b)

Fig. 3. Distribution of random bot inter-message delay (a) and message size (b)



(a) (b)

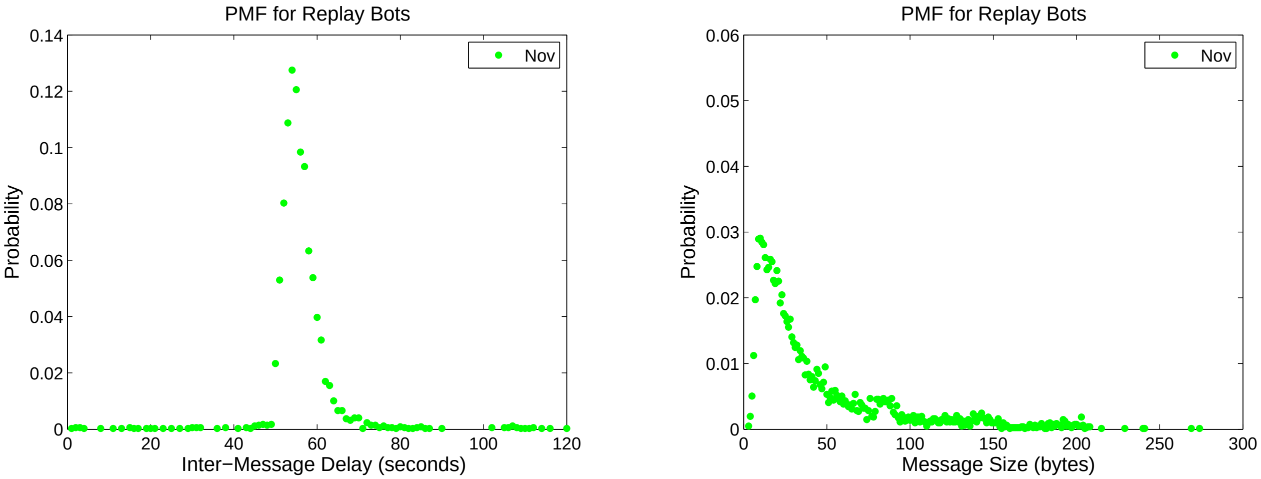
Fig. 4. Distribution of responder bot inter-message delay (a) and message size (b)

addition, responses can be assigned with probabilities, so that the responder bot responds to a given trigger in a random manner.

Figure 4 shows the probability distributions of inter-message delay and message size for responder bots. Note that only the distributions of the August 2007 responder bots are shown due to the small number of responder bots found in November 2007. Since the message emission of responder bots is triggered by human messages, theoretically the distribution of inter-message delays of responder bots should demonstrate certain similarity to that of humans. Figure 4 (a) confirms this hypothesis. Like Figure 1 (a), the PMF of responder bots (excluding the head part) in log-log scale exhibits a clear sign of a heavy tail. But unlike human messages, the sizes of responder bot messages vary in a much narrower range (between 1 and 160). The bell shape of the distribution for message size less than 100 indicates that responder bots share a similar message composition technique with periodic bots, and their messages are composed as templates with multiple parts, as shown in Appendix A.

5) Replay Bots: A replay bot not only sends its own messages, but also repeats messages from other users to appear more like a human user. In our experience, replayed phrases are related to the same topic but do not appear in the same chat room as the original ones. Therefore, replayed phrases are either taken from other chat rooms on the same topic or saved previously in a database and replayed.

The use of replayed phrases in a crowded or "noisy" chat room does, in fact, make replay bots look more like human to inattentive users. The replayed phrases are sometimes nonsensical in the context of the chat, but human users tend to naturally ignore such statements. When replay bots succeed in deceiving human users, these users are more likely to click links posted by the bots or visit their profiles. Interestingly, replay bots sometimes replay phrases uttered by other chat bots, making them very easy to be recognized. The use of replay is potentially effective in thwarting detection methods, as detection tests must deal with a combination of human and bot phrases. By using human phrases, replay bots can easily defeat keyword-based message filters that filter message-by-



(a)

(b)

Fig. 5. Distribution of replay bot inter-message delay (a) and message size (b)

message, as the human phrases should not be filtered out.

Figure 5 illustrates the probability distributions of inter-message delay and message size for replay bots. In terms of inter-message delay, replay bots are just a variation of periodic bots, which is demonstrated by the high spike in Figure 5 (a). By using human phrases, replay bots successfully mimic human users in terms of message size distribution. The message size distribution of replay bots in Figure 5 (b) largely resembles that of human users, and can be fitted by an exponential distribution with $\lambda = 0.028$.

6) Replay-Responder Bots: A natural next step towards statistically human-like bots is to integrate replay and responder bots. A replay-responder bot would respond to user messages based on keyword triggers, like current responder bots, and would also randomly replay human messages, like current replay bots, resulting in human-like inter-message delay and message size statistics. To represent replay-responder bots, we simulate them by combining replay bot messages with responder bot inter-message delays. In addition to message statistics, how often humans are deceived (confuse bots with real humans) by the content of replayed messages, which is detailed in Section III-C, would determine how effective replay-responder bots are for spreading spam or malware.

7) Advanced Responder Bots: The developer of the first-generation responder bot pointed us to a more advanced, next-generation version with a highly detailed configuration, which we refer to as the advanced responder bot. The advanced responder bot is designed to be more human-like by using a large set of keywords and responses. The advanced responder bot has a much larger number of keyword triggers than earlier bots, with each keyword trigger being hand-crafted. The keywords and associated responses are programmed with templated components and random typos, like earlier bots. The developer of the second-generation or advanced responder bot shared its configuration file. The configuration consists of over 11,000 rules. The rules consist of a keyword and its associated set of responses, or a template variable and its associated synonym phrases. By contrast, the configuration files of most basic responder bots have less than 100 rules. Although state-of-the-art AI bots developed for research are based on more technically complicated solutions [35], the advanced responder bots prove very effective at deceiving human users in chat, which is shown in Section III-C.

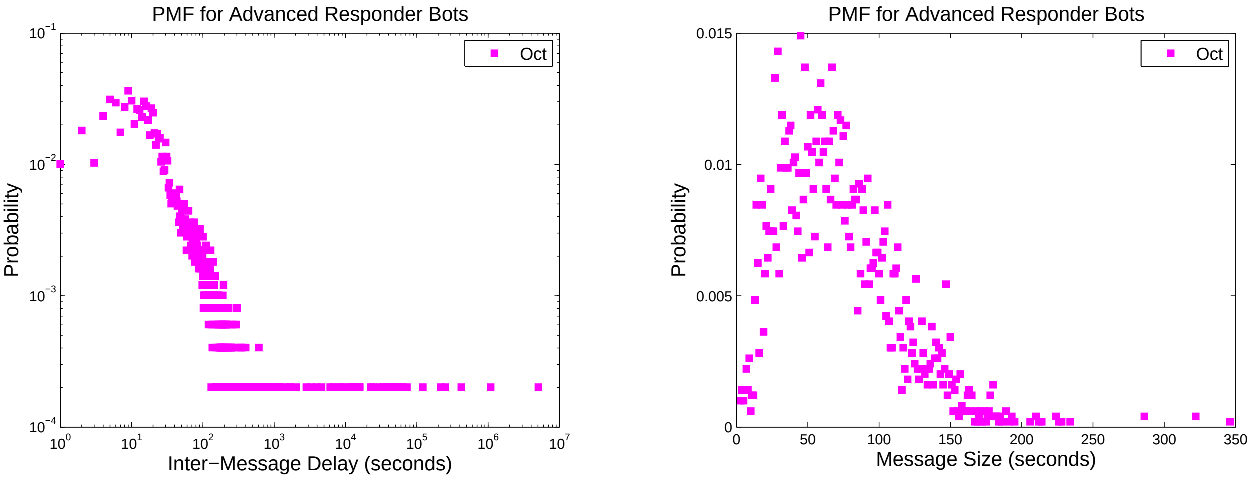
TABLE I
MESSAGE-TO-RESPONSE RATIOS

	message-to-response ratio	
bot type	response from unique users	all responses
periodic	0.0021	0.0024
random	0.0050	0.0072
replay	<0.001	<0.001
responder	0.0097	0.0131
adv. responder	0.0293	0.0957

Figure 6 illustrates the probability distributions of inter-message delay and message size for advanced responder bots. The distribution of inter-message delays for advanced responder bots is much like that of basic responder bots. The message size distribution is also similar, but with a higher proportion of larger messages. Whereas advanced responder bots are similar to basic responder bots in message size and inter-message delay statistics, their more advanced configuration makes their responses and message content more human-like.

C. Conversation Analysis

In this section, we introduce a metric that estimates how often humans respond to bots in chat. In addition to inter-message delay and message size statistics, how bots and humans interact in conversation is important. While our measurements and statistical analysis characterize how human-like bots behave in inter-message delays and message sizes, these statistics do not demonstrate how human-like bots communicate with humans in conversation. The new metric of message-to-response ratio attempts to measure this. It estimates how often humans respond to bots by computing a ratio between the number of messages a bot sends and the number of response messages addressed back that bot. The standard convention in chat rooms for addressing a specific user with a message is to



(a)

(b)

Fig. 6. Distribution of advanced responder bot inter-message delay (a) and message size (b)

add its username before or after the message. For example, to address a message to “alexander_the_great”, a user could start its message with “alex, ” or “alexander, ”; or end its message with “@ alex” or “@ alexander”. This convention is not part of the Yahoo! chat protocol, but is widely used and most chat clients support tab auto-completion of usernames in chat and highlight messages addressed to a user, i.e., messages with text that matches its username.

Note that responses from bots to other bots are not counted and responses from humans to bots telling them to “shut up” or similar are also discarded. The message-to-response ratio is an approximation in that some users do not follow the convention of addressing users by their usernames, and thus, the metric is an estimate of the “true” message-to-response ratio. The message-to-response ratio results are summarized in Table I, which lists the corresponding ratios for all responses and for responses from unique users, i.e., unique respondents, with respect to different types of bots.

1) Periodic Bots: The simplest bots, periodic bots, have very low message-to-response ratios of the different types of bots at 0.21% and 0.24% for responses from unique users and all responses, respectively. This result is not from their periodic timing, but rather due to their general lack of sophistication in message composition, as seen in the synonym template example used by many periodic bots in Appendix B. In general, it appears that less effort has been paid in their development than other bots.

2) Random Bots: The message-to-response ratios of random bots are 0.50% for responses from unique users and 0.72% for all responses, about two to three times higher than those of periodic bots. In most cases, random bots utilize handcrafted messages, as shown in the basic message example used by one of the common random bots in Appendix D. The less robotic messages are better received by humans, increasing the response rate.

3) Responder Bots: Responder bot is another type of advanced chat bot. The response triggering mechanism proves to be very effective at fooling humans. The responder bots in the original dataset, August and November of 2007, achieve a relatively high message-to-response ratio for all responses of 1.31%, about twice that of random bots and six times that of periodic bots, and 0.97% for responses from unique users, implying that some users follow up with multiple responses. By responding to certain keywords, responder bots are able to approximately simulate human conversation.

4) Replay Bots: The most human-like bots in statistical terms are replay and responder bots. Interestingly, despite being human-like in statistical terms, replay bots are incompetent at beguiling humans with the lowest message-to-response ratios, which are <0.001%. Although one might expect that replaying random human messages might be very effective due to the chaos of chat rooms, it proves to be quite the opposite. Instead, most humans see right through this trick and completely ignore these bots.

5) Replay-Responder Bots: The replay-responder bots are simulated and not observed in the wild, so it is impossible to determine their message-to-response ratios. However, due to their message content being the same as replay bots and only their timing being different, it is likely that their message-to-response ratios would be similar to replay bots.

6) Advanced Responder Bots: The advanced responder bots achieve a message-to-response ratio of 2.93% for responses from unique users and 9.57% for all responses. The much higher ratio for all responses indicates that the advanced responder bots often get multiple responses from a same user. There are several reasons for these results. First, conversing in a topic-specific chat room is not equivalent to the very difficult non-restricted Turing test, but rather is more like a restricted Turing test—a much easier variation of the test, because of the room topic. It should also be emphasized that identifying bots in chat rooms is not a conventional Turing test as described in the literature, since users are not expert examiners, and often, are not even aware that bots exist in chat rooms. Second, the advanced responder bots have a much larger configuration file than previous responder bots, about two orders of magnitude larger, as described in Section III-B7.

Third, the advanced responder bots make use of some hard-coded domain knowledge related to the chat room topics. For example, for a chat room on religion, the advanced responder bot is programmed with rules for keywords like "evolution" or "abortion." By doing so, the advanced responder bots are not only able to deceive humans, but often sustain extended conversations with multiple exchanges.

IV. Classification System

This section describes the design of our chat bot classification system. The two main components of our classification system are the entropy classifier and the Bayesian classifier. The basic structure of our chat bot classification system is shown in Figure 7. The two classifiers, entropy and Bayesian, operate concurrently to process input and make classification decisions, while the Bayesian classifier relies on the entropy classifier to build the bot corpus. The entropy classifier uses entropy and corrected conditional entropy to score chat users and then classifies them as chat bots or humans. The main task of the entropy classifier is to capture new chat bots and add them to the chat bot corpus. The human corpus can be taken from a database of clean chat logs or created by manual log-based classification, as described in Section III. The Bayesian classifier uses the bot and human corpora to learn text patterns of bots and humans, and then it can quickly classify chat bots based on these patterns. The two classifiers are detailed as follows.

A. Entropy Classifier

The entropy classifier makes classification decisions based on entropy and entropy rate measures of message sizes and inter-message delays for chat users. If either the entropy or entropy rate is low for these characteristics, it indicates the regular or predictable behavior of a likely chat bot. If both the entropy and entropy rate are high for these characteristics, it indicates the irregular or unpredictable behavior of a possible human.

To use entropy measures for classification, we set a cutoff score for each entropy measure. If a test score is greater than or equal to the cutoff score, the chat user is classified as a human. If the test score is less than the cutoff score, the chat user is classified as a chat bot. The specific cutoff score is an important parameter in determining the false positive and true positive rates of the entropy classifier. On the one hand, if the cutoff score is too high, then too many humans will be misclassified as bots. On the other hand, if the cutoff score is too low, then too many chat bots will be misclassified as humans. Due to the importance of achieving a low false positive rate, we select the cutoff scores based on human entropy scores to achieve a targeted false positive rate. The specific cutoff scores and targeted false positive rates are described in Section IV-B.

1) Entropy Measures: The entropy rate, which is the average entropy per random variable, can be used as a measure of complexity or regularity [38]–[40]. The entropy rate is defined as the conditional entropy of a sequence of infinite length. The entropy rate is upper-bounded by the entropy of the first-order probability density function or first-order entropy. A independent and identically distributed (i.i.d.) process has an entropy rate equal to its first-order entropy. A highly complex process has a high entropy rate, while a highly regular process has a low entropy rate.

A random process $X = \{X_i\}$ is defined as an indexed sequence of random variables. To give the definition of the entropy rate of a random process, we first define the entropy of a sequence of random variables as:

$$H(X_1,\ldots,X_m) = \\ -\sum_{X_1,\ldots,X_m} P(x_1,\ldots,x_m)\log P(x_1,\ldots,x_m),$$

where $P(x_1,\ldots,x_m)$ is the joint probability $P(X_1 = x_1,\ldots, X_m = x_m)$.

Then, from the entropy of a sequence of random variables, we define the conditional entropy of a random variable given a previous sequence of random variables as:

$$H(X_m \mid X_1,\ldots,X_{m-1}) = \\ H(X_1,\ldots,X_m) - H(X_1,\ldots,X_{m-1}).$$

Lastly, the entropy rate of a random process is defined as:

$$\overline{H}(X) = \lim_{m\to\infty} H(X_m \mid X_1,\ldots,X_{m-1}).$$

Theoretically, since the entropy rate is the conditional entropy of a sequence of infinite length, it cannot be measured for finite samples. Thus, we estimate the entropy rate with the conditional entropy of finite samples. In practice, we replace probability density functions with empirical probability density functions based on the method of histograms. The data is binned in Q bins of approximately equal probability. The empirical probability density functions are determined by the proportions of bin number sequences in the data, i.e., the proportion of a sequence is the probability of that sequence. The estimates of the entropy and conditional entropy, based on empirical probability density functions, are represented as: EN and CE, respectively.

There is a problem with the estimation of $CE(X_m \mid X_1,\ldots,X_{m-1})$ for some values of m. The conditional entropy tends to zero as m increases, due to limited data. If a specific sequence of length $m-1$ is found only once in the data, then the extension of this sequence to length m will also be found only once. Therefore, the length m sequence can be predicted by the length $m-1$ sequence, and the length m and $m-1$ sequences cancel out. If no sequence of length m is repeated in the data, then $CE(X_m \mid X_1,\ldots,X_{m-1})$ is zero, even for i.i.d. processes.

To solve the problem of limited data, without fixing the length of m, we use the corrected conditional entropy [38] represented as CCE. The corrected conditional entropy is defined as:

$$CCE(X_m \mid X_1,\ldots,X_{m-1}) = \\ CE(X_m \mid X_1,\ldots,X_{m-1}) + perc(X_m)\cdot EN(X_1),$$

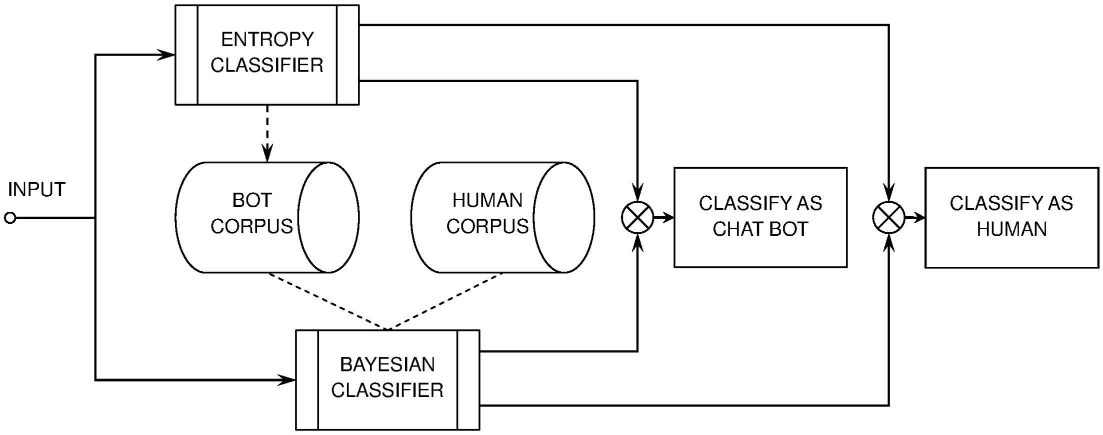


Fig. 7. Classification System Diagram

where $perc(X_m)$ is the percentage of unique sequences of length m and $EN(X_1)$ is the entropy with m fixed at 1 or the first-order entropy.

The estimate of the entropy rate is the minimum of the corrected conditional entropy over different values of m. The minimum of the corrected conditional entropy is considered to be the best estimate of the entropy rate from the available data.

B. Bayesian Classifier

The Bayesian classifier uses the content of chat messages to identify chat bots. Since chat messages (including emoticons) are text, the identification of chat bots can be perfectly fitted into the domain of Bayesian text classification. Within the Bayesian paradigm, the text classification problem can be formalized as $f : T \times C \rightarrow \{0,1\}$, where f is the classifier, $T = \{t_1,t_2,...,t_n\}$ is the texts to be classified, and $C = \{c_1,c_2,...,c_k\}$ is the set of pre-defined classes [41]. Value 1 for $f(t_i,c_j)$ indicates that text t_i is in class c_j and value 0 indicates the opposite decision. There are many techniques that can be used for text classification, such as naïve Bayes, support vector machines, and decision trees. Among them, Bayesian classifiers have been very successful in text classification, particularly in email spam detection. Due to the similarity between chat spam and email spam, we choose Bayesian classification for our text classifier for detecting chat bots. We leave study on the applicability of other types of text classifiers to our future work.

Within the framework of Bayesian classification, identifying if chat message M is issued by a bot or human is achieved by computing the probability of M being from a bot with the given message content, i.e., $P(C = bot|M)$. If the probability is equal to or greater than a pre-defined threshold, then message M is classified as a bot message. According to Bayes theorem,

$$P(bot|M) = \frac{P(M|bot)P(bot)}{P(M)} = \frac{P(M|bot)P(bot)}{P(M|bot)P(bot)+P(M|human)P(human)}.$$

A message M is described by its feature vector $\langle f_1, f_2,..., f_n\rangle$. A feature f is a single word or a combination of multiple words in the message. To simplify computation, in practice it is usually assumed that all features are conditionally independent with each other for the given category. Thus, we have

$$P(bot|M) = \frac{P(bot)\prod_{i=1}^{n} P(f_i|bot)}{P(bot)\prod_{i=1}^{n} P(f_i|bot)+P(human)\prod_{i=1}^{n} P(f_i|human)}.$$

The value of $P(bot|M)$ may vary in different implementations (see [26] for implementation details) of Bayesian classification due to differences in assumption and simplification.

Given the abundance of available implementations of Bayesian classification, we directly adopt an existing implementation, namely the CRM114 Descriminator [24], as our Bayesian classification component. CRM114 is a powerful text classification system that has achieved very high accuracy in email spam identification. The default classifier of CRM114, OSB (Orthogonal Sparse Bigram), is a type of Bayesian classifier. Different from common Bayesian classifiers which treat individual words as features, OSB uses word pairs as features instead. OSB first chops the whole input into multiple basic units with five consecutive words in each unit. Then, it extracts four word pairs from each unit to construct features, and derives their probabilities. Finally, OSB applies Bayes theorem to compute the overall probability that the text belongs to one class or another.

sectionExperimental Evaluation

In this section, we evaluate the effectiveness of our proposed classification system. Our classification tests are based on chat logs collected from the Yahoo! chat system. We test the two classifiers, entropy-based and Bayesian-based, against chat bots from August and November datasets. The Bayesian classifier is tested with fully-supervised training and entropy-classifier-based training. The accuracy of classification is measured in terms of false positive and false negative rates against the labeled datasets, which are used as the ground truth of this work as described in Section III-A. The false positives are those human users that are misclassified as chat bots, while the false negatives are those chat bots that are misclassified as

TABLE II
MESSAGE COMPOSITION OF CHAT BOT AND HUMAN DATASETS

	AUG. BOTS			NOV. BOTS			ADV. BOTS		HUMANS
	periodic	random	responder	periodic	random	replay	replay-responder	adv. responder	human
number of messages	25,258	13,998	6,160	10,639	22,820	8,054	6,160	4,975	342,696

human users. The speed of classification is mainly determined by the minimum number of messages that are required for accurate classification. In general, a high number means slow classification, whereas a low number means fast classification.

C. Experimental Setup

The chat logs used in our experiments are mainly in four datasets: (1) human chat logs from August 2007, (2) bot chat logs from August 2007, (3) bot chat logs from November 2007, and (4) bot chat logs from October 2008. In total, these chat logs contain 342,696 human messages and 92,049 bot messages. In our experiments, we use the first half of each chat log, human and bot, for training our classifiers and the second half for testing our classifiers. The composition of the chat logs for the four datasets is listed in Table II.

The entropy classifier only requires a human training set. We use the human training set to determine the cutoff scores, which are used by the entropy classifier to decide whether a test sample is a human or bot. The target false positive rate is set at 0.01. To achieve this false positive rate, the cutoff scores are set at approximately the 1st percentile of human training set scores. Then, samples that score higher than the cutoff are classified as humans, while samples that score lower than the cutoff are classified as bots. The entropy classifier uses two entropy tests: entropy and corrected conditional entropy. The entropy test estimates first-order entropy, and the corrected conditional entropy estimates higher-order entropy or entropy rate. The corrected conditional entropy test is more precise with coarse-grain bins, whereas the entropy test is more accurate with fine-grains bins [40]. Therefore, we use $Q = 5$ for the corrected conditional entropy test and $Q = 256$ with m fixed at 1 for the entropy test.

We run classification tests for each bot type using the entropy classifier and the Bayesian classifier. The Bayesian classifier is tested based on the fully-supervised training and the entropy-based training. In the fully-supervised training, the Bayesian classifier is trained with manually labeled data, as described in Section III. In the entropy-based training, the Bayesian classifier is trained with data labeled by the entropy classifier. For each evaluation, the entropy classifier uses samples of 100 messages and the Bayesian classifier uses samples of 25 messages, except where noted otherwise.

D. Experimental Results

We now present the detection results for the entropy classifier and the Bayesian classifier. The classification tests and corresponding results are organized by chat bot type, and are ordered by increasing detection difficulty—periodic, random, responder, replay, advanced responder, and replay-responder. After the bot-related results, the human results are presented.

TABLE IV
ENTROPY CLASSIFIER ACCURACY FOR ADVANCED BOTS

	ADV. BOTS	
	replay-responder	adv. responder
test	true pos.	true pos.
EN-imd	3% (1/30)	0% (0/24)
CCE-imd	13% (4/30)	4% 1/24
EN-ms	0% (0/30)	8% (2/24)
CCE-ms	0% (0/30)	91% (22/24)
OVERALL	13% (4/30)	91% (22/24)

1) Entropy Classifier: The detection results of the entropy classifier are listed in Tables III and IV, which include the results of the entropy test (EN) and corrected conditional entropy test (*CCE*) for inter-message delay (*imd*), and message size (*ms*). For each type of bot, we first present the EN-imd and CCE-imd results and then the EN-ms and CCE-ms results. The overall results for all entropy-based tests are shown in the final row of the table. The true positives (shown as true positives over the total number of bots) are the bot samples correctly classified as bots. The false positives (shown as false positives over the total number of humans) are the human samples mistakenly classified as bots.

Periodic Bots: As the simplest group of bots, periodic bots are the easiest to detect. They use different fixed timers and repeatedly post messages at regular intervals. Therefore, their inter-message delays are concentrated in a narrower range than those of humans, resulting in lower entropy than that of humans. The EN-imd and CCE-imd tests detect 100% of all periodic bots in both August and November datasets. The EN-ms and CCE-ms tests detect 76% and 63% of the August periodic bots, respectively, and 90% and 100% of the November periodic bots, respectively. These slightly lower detection rates are due to a small proportion of humans with low entropy scores that overlap with some periodic bots. These humans post mainly short messages, resulting in message size distributions with low entropy.

Random Bots: The random bots use random timers with different distributions. Some random bots use discrete timings, e.g., 40, 64, or 88 seconds, while the others use continuous timings, e.g., uniformly distributed delays between 45 and 125 seconds.

The EN-imd and CCE-imd tests detect 100% of all random bots, with one exception: the CCE-imd test against the August random bots only achieves 72% detection rate, which is caused by the following two conditions: (1) the range of message delays of random bots is close to that of humans; (2) sometimes the randomly-generated delay sequences have similar entropy rate to human patterns. The EN-ms and CCE-ms tests detect 10% and 11% of August random bots, respectively, and 31% and 5% of November random bots, respectively.

TABLE III
Entropy Classifier Accuracy

	AUG. BOTS			NOV. BOTS			HUMANS
	periodic	random	responder	periodic	random	replay	human
test	true pos.	true pos.	true pos.	true pos.	true pos.	true pos.	false pos.
EN-imd	100% (121/121)	100% (68/68)	3% (1/30)	100% (51/51)	100% (109/109)	100% (40/40)	<1% (7/1713)
CCE-imd	100% (121/121)	72% (49/68)	13% (4/30)	100% (51/51)	100% (109/109)	100% (40/40)	<1% (11/1713)
EN-ms	76% (92/121)	10% (7/68)	26% (8/30)	90% (46/51)	31% (34/109)	0% (0/40)	<1% (7/1713)
CCE-ms	63% (77/121)	11% (8/68)	100% (30/30)	100% (51/51)	5% (6/109)	0% (0/40)	<1% (11/1713)
OVERALL	100% (121/121)	100% (68/68)	100% (30/30)	100% (51/51)	100% (109/109)	100% (40/40)	<1% (17/1713)

TABLE V
Bayesian Classifier Accuracy

	AUG. BOTS			NOV. BOTS			HUMANS
	periodic	random	responder	periodic	random	replay	human
test	true pos.	true pos.	true pos.	true pos.	true pos.	true pos.	false pos.
SupBC	100% (121/121)	100% (68/68)	100% (30/30)	27% (14/51)	95% (104/109)	2% (1/40)	0% (0/1713)
SupBCretrained	100% (121/121)	100% (68/68)	100% (30/30)	100% (51/51)	100% (109/109)	100% (40/40)	0% (0/1713)
EntBC	100% (121/121)	100% (68/68)	100% (30/30)	100% (51/51)	100% (109/109)	100% (40/40)	<1% (1/1713)

These low detection rates are again due to a small proportion of humans with low message size entropy scores. However, unlike periodic bots, the message size distribution of random bots is highly dispersed, and thus, a larger proportion of random bots have high entropy scores, which overlap with those of humans.

Responder Bots: The responder bots are among the advanced bots, and they behave more like humans than random or periodic bots. They are triggered to post messages by certain human phrases. As a result, their timings are quite similar to those of humans.

The EN-imd and CCE-imd tests detect very few responder bots, only 3% and 13%, respectively. This demonstrates that human-message-triggered responding is a simple yet very effective mechanism for imitating the timing of human interactions. However, the detection rate for the EN-ms test is slightly better at 26%, and the detection rate for the CCE-ms test reaches 100%. While the message size distribution has sufficiently high entropy to frequently evade the EN-ms tests, there is some dependence between subsequent message sizes, and thus, the CCE-ms detects the low entropy pattern over time.

Replay Bots: The replay bots also belong to the advanced and human-like bots. They use replay attacks to fool humans. More specifically, the bots replay phrases they observed in chat rooms. Although not sophisticated in terms of implementation, the replay bots are quite effective in deceiving humans as well as frustrating our message-size-based detections: the EN-ms and CCE-ms tests both have detection rates of 0%. Despite their clever trick, the timing of replay bots is periodic and easily detected. The EN-imd and CCE-imd tests are very successful at detecting replay bots, both with 100% detection accuracy.

Replay-Responder Bots: The replay-responder bot is a simulated hybrid of the two advanced bot types: replay and responder bots. By integrating replay bot message size with responder bot timing, these bots are effective in capturing human-like timing and message-size statistics. The replay-responder bots share the replay bots' effectiveness in defeating message-size-based detection: the EN-ms and CCE-ms tests both detect 0% of replay-responder bots. The timing of replay-responder bots is also human-like, inherited from responder bots. The detection rate of the EN-imd test is only 3% and that of the inter-message delay CCE-imd test is 13%.

TABLE VI
Bayesian Classifier Accuracy for Advanced Bots

	ADV. BOTS	
	replay-responder	adv. responder
test	true pos.	true pos.
SupBC	3% (1/30)	0% (0/24)
SupBCretrained	100% (30/30)	100% (24/24)
EntBC - 25 msgs.	83% (25/30)	100% (24/24)
EntBC - 50 msgs.	93% (28/30)	100% (24/24)
EntBC - 75 msgs.	96% (29/30)	100% (24/24)
EntBC - 100 msgs.	100% (30/30)	100% (24/24)

Advanced Responder Bots: The advanced responder bots, as discussed in Section III-B7, are a highly customized version of regular responder bots and are especially effective at engaging and interacting with human users in chat, as shown in Section III-C6. The advanced responder bot, like its basic version, is insusceptible to those tests based on inter-message delay. The detection rates of the EN-imd and CCE-imd tests are 0% and 4%, respectively. Also like the basic version, the advanced responder bots is sensitive to the tests based on message size, with the detection rate of the EN-ms test at 8% and that of the CCE-ms test at 91%.

Humans: A few humans fall under the cutoffs for the EN and CCE tests, resulting in false positives, i.e., humans misclassified as bots. These misclassified humans also do not have clearly repeated patterns in the timing or size of their messages. However, their variations are not as high as those of correctly classified humans, resulting in low entropies for size and inter-message delay.

2) Supervised and Hybrid Bayesian Classifiers: The detection results of the Bayesian classifier are listed in Tables V and VI. Here the fully-supervised Bayesian classifier and entropy-trained Bayesian classifier, both trained on the August training datasets, are represented as *SupBC* and *EntBC*, respectively; while the fully-supervised Bayesian classifier trained on August and November training datasets is represented as

SupBCretrained.

Periodic Bots: For the August dataset, both the *SupBC* and *EntBC* classifiers detect 100% of all periodic bots. For the November dataset, however, the *SupBC* classifier only detects 27% of all periodic bots. The lower detection rate is due to the fact that 62% of the periodic bot messages in November chat logs are generated by new bots, making the *SupBC* classifier ineffective without re-training. The *SupBCretrained* classifier detects 100% of November periodic bots. The *EntBC* classifier also achieves 100% for the November dataset.

Random Bots: For the August dataset, both the *SupBC* and *EntBC* classifiers detect all the random bots. For the November dataset, the *SupBC* classifier detects 95% of the random bots, and the *SupBCretrained* classifier detects 100% of the random bots. While 52% of the random bots have been upgraded according to our observation, the old training set is still mostly effective. This is because certain content features of August random bots still appear in November. The *EntBC* classifier again achieves 100% detection accuracy for the November dataset.

Responder Bots: We only present the detection results of the responder bots in the August dataset, as the number of the responder bots in the November dataset is very small. Although responder bots effectively mimic human timing, their message contents are only slightly obfuscated and can be easily detected. The *SupBC* and *EntBC* classifiers both detect all the responder bots.

Replay Bots: The replay bots only exist in the November dataset. The *SupBC* classifier detects only 2% of the replay bots, as these bots are newly introduced in November. However, the *SupBCretrained* classifier detects 100% of the replay bots. The Bayesian classifier reliably detects the replay bots in the presence of a substantial number of replayed human phrases, indicating the effectiveness of Bayesian techniques in chat bot classification.

Replay-Responder Bots: The *SupBC* results for replay-responder bots are only 3%, similar to those for basic replay bots. The *SupBCretrained* classifier again detects 100% of replay-responder bots after being trained on their message content. Whereas overall entropy is only able to detect 13% of replay-responder bots, *EntBC* is able to detect 83% with 25 messages and up to 100% when the *EntBC* classifier uses 100 messages for its decisions. Interestingly, these results show that a classifier with both low true positive and false positive rates can still be useful for training another classifier and that trained classifier can be highly accurate, i.e., high true positive and low false positive rates. The content features of replay-responder bots are learned, resulting in both higher true positive and lower false positive rates for *EntBC* than the entropy classifiers achieved. The correctly-labeled replay-responder bot messages used by *EntBC* for training contain distinct content patterns. The wrongly-labeled human messages used by *EntBC* for training are unique, random human phrases with no apparent pattern, their only common feature being low entropy in inter-message delay or message size statistics. Thus, the *EntBC* classifier effectively learns to recognize the repeated patterns in the replay-responder bot messages, while the occasional wrongly-labeled human message has little affect on the training.

Advanced Responder Bots: Although these bots are more advanced than regular responder bots, it is not more difficult to detect them through entropy or Bayesian classification. The *EntBC* classifier detects all the advanced responder bots. The *SupBC* and *SupBCretrained* classifiers both fails to detect the advanced responder bots, due to being only trained on August and November 2007 datasets. However, after being trained on the October 2008 training set, *SupBCretrained* detects 100% of the advanced responder bots.

Humans: The *SupBC* and *SupBCretrained* classifiers correctly identify all of the humans in the dataset. Interestingly, *EntBC* is more accurate than the original entropy classifier, only having one false positive. Note that the wrongly-classified human messages by the entropy classifier are only a few and are basically random human phrases with no obvious content patterns. Because such random phrases are not likely to be repeated, even less humans are misclassified by *EntBC* than the entropy classifier.

V. Conclusion and Future Work

This paper first presents a large-scale measurement study on Internet chat. We collected two-month chat logs for 21 different chat rooms from one of the top Internet chat service providers. From the chat logs, we identified a total of 16 different types of chat bots and grouped them into six categories: periodic bots, random bots, responder bots, replay bots, replay-responder bots, and advanced responder bots. Through statistical analysis on inter-message delay and message size for both chat bots and humans, we found that chat bots behave very differently from human users. More specifically, chat bots exhibit certain regularities in either inter-message delay or message size. Although responder bots and replay bots employ advanced techniques to behave more human-like in some aspects, they still lack the overall sophistication of humans.

Based on the measurement study, we further proposed a chat bot classification system, which utilizes entropy-based and Bayesian-based classifiers to accurately detect chat bots. The entropy-based classifier exploits the low entropy characteristic of chat bots in either inter-message delay or message size, while the Bayesian-based classifier leverages the message content difference between humans and chat bots. The entropy-based classifier is able to detect unknown bots, including human-like bots such as responder and replay bots. However, it takes a relatively long time for detection, i.e., a large number of messages are required. Compared to the entropy-based classifier, the Bayesian-based classifier is much faster, i.e., a small number of messages are required. In addition to bot detection, a major task of the entropy-based classifier is to build and maintain the bot corpus. With the help of bot corpus, the Bayesian-based classifier is trained, and consequently, is able to detect chat bots quickly and accurately. Our experimental results demonstrate that the hybrid classification system is fast in detecting known bots and is accurate in identifying previously-unknown bots.

There are a number of possible areas for future work. In particular, practical deployment would raise several questions.

While our current system was trained on data collected over half a month, the same volume of data could be collected in only a few hours system wide. With a large volume of data, the system could be retrained quite often and old training data would need to be aged out. Although aging methods used for spam filtering such as microgrooming [24] and exponential aging [42] are applicable for this study, further research is needed to determine the best approach.

We also plan to investigate more advanced chat bots. For example, multiple bots could collude to forge real conversations or could perform relay attacks [43] to exploit vulnerable human users. We believe that continued work in this area will reveal other important characteristics of bots and automated programs, which could be useful in malware detection and prevention.

Appendix

Note that in the examples that follow, the messages would be spread out over several minutes and interleaved with messages from other users.

A. Response Example

In the following example, the bot responds to keywords using a template with three parts to post its response messages and links: *[username], [link description phrase]; [link].*

```
bot: user1, that’s a damn good question.
bot: user1, To know more about Seventh-day
Adventist; visit http://www.sda.org
bot: user2, no! don’t leave me.
bot: user1, too much coffee tonight?
bot: user2, boy, you’re just full of
questions, aren’t you?
bot: user2, lots of evidence
for evolution can be found here
http://www.talkorigins.org/faqs/comdesc/
```

B. Synonym Example

In the following example, the bot uses a template with three parts to post messages: *[salutation phrase]! [introduction phrase]! [web site advertisement phrase].*

```
bot: Allo Hunks! Enjoy Marjorie! Check My Free
Pics
bot: What’s happening Guys! Marjorie Here! See
more of me at My Free Pics
bot: Hi Babes! I am Marjorie! Rate My Live Cam
bot: Horny lover Guys! Marjorie at your
service! Inspect My Site
bot: Mmmm Folks! Im Marjorie! View My Webpage
```

C. Padding Example

In the following example, the bot adds random characters to messages.

```
bot: anyone boredjn wanna chat?uklcss
bot: any guystfrom the US/Canada hereiqjss
bot: hiyafxqss
bot: ne1 hereqbored?fiqss
bot: ne guysmwanna chat? ciuneed some1 to make
megsmile :-)pktpss
```

D. Hand-Crafted Example

In the following example, the bot uses verbatim hand-crafted messages from a small database, i.e., no templates.

```
bot: pm me guys, I’m bored.
bot: guys? PM ME!
bot: what are you guys doing, someone make
this girl smile!
bot: helloooo people! someone chat with me I’m
so bored
bot: what’s everyone up to?
```

Acknowledgments

The authors would like to thank the anonymous reviewers for their detailed and insightful comments. This work was partially supported by NSF grant 0901537 and ONR grant N00014-09-1-0746.

References

[1] C. Dewes, A. Wichmann, and A. Feldmann, “An analysis of Internet chat systems,” in *Proceedings of IMC 2003*, Miami, FL., USA, October 2003.

[2] J. Hu, “AOL spam petitions cut both ways,” http://news.cnet.com/AOL-spam-petitions-cut-both-ways/2100-1024_3-1015385.html [Accessed: Dec. 22, 2010].

[3] B. Krebs, “Yahoo! messenger network overrun by bots,” http://blog.washingtonpost.com/securityfix/2007/08/yahoo_messenger_network_overru.html [Accessed: Dec. 18, 2007].

[4] A. Mohta, “Yahoo! chat adds CAPTCHA check to remove bots,” http://www.technospot.net/blogs/yahoo-chat-captcha-check-to-remove-bots/ [Accessed: Dec. 18, 2007].

[5] ——, “Bots are back in Yahoo! chat rooms,” http://www.technospot.net/blogs/bots-are-back-in-yahoo-chat-room/ [Accessed: Dec. 18, 2007].

[6] V. Hinze-Hoare, “Should cyberspace chat rooms be closed to protect children?” *Computing Research Repository*, 2004.

[7] S. Bono, D. Caselden, G. Landau, and C. Miller, “Reducing the attack surface in massively multiplayer online role-playing games,” *IEEE Security and Privacy*, vol. 7, 2009.

[8] Yahelite.org, “Yahelite chat client,” http://www.yahelite.org/ [Accessed: Jan. 8, 2008].

[9] L. V. Ahn, M. Blum, N. Hopper, and J. Langford, “CAPTCHA: Using hard AI problems for security,” in *Proceedings of Eurocrypt*, Warsaw, Poland, May 2003.

[10] R. B. Jennings III, E. M. Nahum, D. P. Olshefski, D. Saha, Z.-Y. Shae, and C. Waters, “A study of internet instant messaging and chat protocols,” *IEEE Network*, vol. 20, no. 4, pp. 16–21, 2006.

[11] E. Mills, “Yahoo! closes chat rooms over child sex concerns,” http://news.cnet.com/2100-1025_3-5759705.html [Accessed: Jan. 27, 2008].

[12] Z. Liu, W. Lin, N. Li, and D. Lee, “Detecting and filtering instant messaging spam - a global and personalized approach,” in *Proceedings of NPSEC 2005*, Boston, MA., USA, November 2005.

[13] M. Mannan and P. C. van Oorschot, “On instant messaging worms, analysis and countermeasures,” in *Proceedings of the ACM Workshop on Rapid Malcode*, Fairfax, VA., USA, November 2005.

[14] M. Xie, Z. Wu, and H. Wang, “HoneyIM: Fast detection and suppression of instant messaging malware in enterprise-like networks,” in *Proceedings of ACSAC 2007*, Miami Beach, FL., USA, December 2007.

[15] A. J. Trivedi, P. Q. Judge, and S. Krasser, “Analyzing network and content characteristics of spim using honeypots,” in *Proceedings of USENIX SRUTI 2007*, Santa Clara, CA., USA, June 2007.

[16] P. J. Y. Jonathan, C. C. Fung, and K. W. Wong, “Devious chatbots - interactive malware with a plot,” *Communications in Computer and Information Science*, vol. 44, 2009.

[17] J. Goebel and T. Holz, “Rishi: Identify bot contaminated hosts by IRC nickname evaluation,” in *Proceedings of HotBots 2007*, Cambridge, MA., USA, April 2007.

[18] D. Dagon, G. Gu, C. P. Lee, and W. Lee, “A taxonomy of botnet structures,” in *Proceedings of ACSAC 2007*, Miami, FL., USA, December 2007.

[19] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, “Bothunter: Detecting malware infection through IDS-driven dialog correlation,” in *Proceedings of USENIX Security 2007*, Boston, MA., USA, August 2007.

[20] G. Gu, J. Zhang, and W. Lee, “BotSniffer: Detecting botnet command and control channels in network traffic,” in *Proceedings of NDSS 2008*, San Diego, CA., USA, February 2008.

[21] J. Nazario. (2007, February-March) Botnet Tracking: Tools, Techniques, and Lessons Learned. Washington, DC., USA.

[22] B. B. Kang, E. Chan-Tin, C. P. Lee, J. Tyra, H. J. Kang, C. Nunnery, Z. Wadler, G. Sinclair, N. Hopper, D. Dagon, and Y. Kim, “Towards complete node enumeration in a peer-to-peer botnet,” in *Proceedings of ASIACCS 2009*, Sydney, Australia, 2009.

[23] S. Nagaraja, P. Mittal, C.-y. Hong, M. Caesar, and N. Borisov, “BotGrep: Finding P2P Bots with Structured Graph Analysis,” in *USENIX Security Symposium 2010*, Washington, DC., USA, August 2010.

[24] W. Yerazunis, “Sparse binary polynomial hashing and the CRM114 discriminator,” in *Proceedings of MIT Spam Conference*, Cambridge, MA., USA, January 2003.

[25] J. Blosser and D. Josephsen, “Scalable centralized bayesian spam mitigation with bogofilter,” in *Proceedings of USENIX LISA 2004*, Atlanta, GA., USA, November 2004.

[26] J. A. Zdziarski, *Ending Spam: Bayesian Content Filtering and the Art of Statistical Language Classification*. No Starch Press, 2005.

[27] K. Li and Z. Zhong, “Fast statistical spam filter by approximate classifications,” in *Proceedings of ACM SIGMETRICS 2006*, St. Malo, France, June 2006.

[28] G. L. Wittel and S. F. Wu, “On attacking statistical spam filters,” Mountain View, CA., USA, July 2004.

[29] D. Lowd and C. Meek, “Good word attacks on statistical spam filters,” Mountain View, CA., USA, July 2005.

[30] C. Karlberger, G. Bayler, C. Kruegel, and E. Kirda, “Exploiting redundancy in natural language to penetrate bayesian spam filters,” in *Proceedings of USENIX WOOT 2007*, Boston, MA., USA, August 2007.

[31] S. Bacon, “New entry process for chat rooms,” http://www.ymessengerblog.com/blog/2007/08/29/new-entry-process-for-chat-rooms/ [Accessed: Jan. 25, 2008].

[32] ——, “Chat rooms follow-up,” http://www.ymessengerblog.com/blog/2007/08/21/chat-rooms-follow-up/ [Accessed: Jan. 25, 2008].

[33] ——, “Chat rooms update,” http://www.ymessengerblog.com/blog/2007/08/24/chat-rooms-update-2/ [Accessed: Jan. 25, 2008].

[34] A. M. Turing, “Computing machinery and intelligence,” *Mind*, vol. 59, pp. 433–460, 1950.

[35] L. Floridi, M. Taddeo, and M. Turilli, “Turing’s imitation game: Still an impossible challenge for all machines and some judges – an evaluation of the 2008 Loebner contest,” *Minds and Machines*, vol. 19, no. 1, 2009.

[36] Symantec Security Response, “W32.Imaut.AS worm,” http://www.symantec.com/security_response/writeup.jsp?docid=2007-080114-2713-99 [Accessed: Jan. 25, 2008].

[37] Uber-Geek.com, “Yahoo! responder bot,” http://www.uber-geek.com/bot.html [Accessed: Jan. 18, 2008].

[38] A. Porta, G. Baselli, D. Liberati, N. Montano, C. Cogliati, T. Gnecchi-Ruscone, A. Malliani, and S. Cerutti, “Measuring regularity by means of a corrected conditional entropy in sympathetic outflow,” *Biological Cybernetics*, vol. 78, no. 1, January 1998.

[39] R. Rosipal, “Kernel-based regression and objective nonlinear measures to assess brain functioning,” Ph.D. dissertation, University of Paisley, Paisley, Scotland, UK, September 2001.

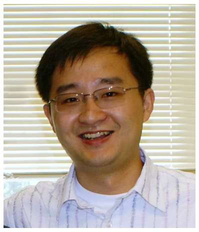
[40] S. Gianvecchio and H. Wang, “Detecting covert timing channels: An entropy-based approach,” in *Proceedings of ACM CCS 2007*, Alexandria, VA., USA, October 2007.

[41] F. Sebastiani, “Machine learning in automated text categorization,” *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[42] Y. Zhou, M. S. Mulekar, and P. Nerellapalli, “Adaptive Spam Filtering Using Dynamic Feature Space,” in *Proceedings of the 2005 International Conference on Tools with Artificial Intelligence*, Hong Kong, China, November 2005, pp. 302–309.

[43] T. Lauinger, V. Pankakoski, and E. Kirda, “Honeybot, your man in the middle for automated social engineering,” in *Proceedings of the USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET’10)*, April 2010.

Steven Gianvecchio received his Ph.D. in Computer Science from the College of William and Mary in 2010. He is a Senior Scientist at the MITRE Corporation, McLean, VA. His research interests include networks, distributed systems, network monitoring, intrusion detection, traffic modeling, and covert channels.



Zhenyu Wu received his M.Sc. degree in Computer Science from the College of William and Mary in 2005. He is currently a Ph.D. candidate in Computer Science at the College of William and Mary. His current research area focuses on data center resource management and network optimization. His research interest also lies in system and network security, including but not limited to malware analysis, packet filters, and Internet chat and online game security.

Mengjun Xie received the Ph.D. degree in Computer Science from College of William and Mary, Williamsburg, in 2009. He is an Assistant Professor of Computer Science at University of Arkansas at Little Rock, Little Rock. His research interests include network security, information security, network systems, and operating systems. He is a member of IEEE.

Haining Wang received his Ph.D. in Computer Science and Engineering from the University of Michigan at Ann Arbor in 2003. He is an Associate Professor of Computer Science at the College of William and Mary, Williamsburg, VA. His research interests lie in the area of networking systems, security, and distributed computing. He is a senior member of IEEE.