情報処理学会論文誌 Vol.54 No.3 1087–1098 (Mar. 2013)

機械的通信挙動モデルに基づく階層型クラスタリングによるボット検知手法

溝口 誠一郎[1,2,a)] 笠原 義晃[2,3,b)] 堀 良彰[1,2,c)] 櫻井 幸一[1,2,d)]

受付日 2012年6月29日, 採録日 2012年12月7日

概要：ネットワーク上に存在するボットに感染した端末を特定するために，ネットワークアプリケーションが送信するアプリケーションプロトコルメッセージの送信間隔に着目したボット検知手法を提案する．人間がアプリケーションを操作した場合，そのアプリケーションプロトコルメッセージの送信間隔はばらつくのに対し，ボットの場合はその挙動がコードによって規定されているため，アプリケーションプロトコルメッセージの送信間隔の分布に偏りが起きる．この分布の違いを利用して，人間とそうでないものを区別することで，ボットを発見する．はじめにネットワークアプリケーションに対する入力と出力の関係をモデル化し，IRC クライアントの IRC メッセージ送信間隔のモデル化を行う．続いて，提案モデルに従ったボット検知アルゴリズムを設計する．アルゴリズムでは，IRC メッセージの送信間隔の列に対して階層型クラスタリングを適用することで人間と機械のモデルの区別を行う．評価では，モデルに基づく擬似データを用いてアルゴリズムのパラメータを設定した後，実際の人間が操作する IRC クライアントで観測された IRC トラフィックならびに IRC ボットのトラフィックに対して提案手法を適用した．その結果，IRC ボットのトラフィックは機械が生成したトラフィックとして正しく判定された．

キーワード：ボット検知，アプリケーションプロトコルメッセージ送信間隔，機械的通信挙動モデル，IRC，階層型クラスタリング

A Bot Detection Method Using Hierarchical Clustering Based on Mechanical Communication Behavior Model

Seiichiro Mizoguchi[1,2,a)] Yoshiaki Kasahara[2,3,b)] Yoshiaki Hori[1,2,c)] Kouichi Sakurai[1,2,d)]

Received: June 29, 2012, Accepted: December 7, 2012

Abstract: In this paper, we propose a bot detection method which focuses on application protocol message transmission intervals of applications in order to find bot infected machine on a network. Our method predicts who is controlling the application by monitoring its network behavior, especially application protocol message transmission intervals. An application which is operated by a human has random behaviors due to the human operation, while a bot has a mechanical behavior since its behavior is written in its own code. First, we build a model of network behavior of human and non-human operated application and we find that several samples follow the model. Then we design a bot detection algorithm using a hierarchical clustering. In evaluation phase, we set parameters in the algorithm with artificial data based on our proposed model and then we evaluate our method with real human IRC traffic and IRC bot traffic. Our method correctly judges bot traffic as machine-generated one.

Keywords: bot detection, application protocol message transmission intervals, mechanical communication behavior model, IRC, hierarchical clustering

1. はじめに

1.1 背景

インターネット上には，分散サービス不能攻撃やフィッ

[1] 九州大学大学院システム情報科学府
Information Science and Electrical Engineering, Kyushu University, Nishi, Fukuoka 819–0395, Japan
[2] 財団法人九州先端科学技術研究所
Institute of Systems, Information Technologies and Nanotechnologies, Fukuoka 814–0001, Japan
[3] 九州大学情報基盤研究開発センター
Research Institute for Information Technology, Kyushu University, Higashi, Fukuoka 812–8581, Japan

a) mizoguchi@itslab.inf.kyushu-u.ac.jp
b) kasahara@nc.kyushu-u.ac.jp
c) hori@inf.kyushu-u.ac.jp
d) sakurai@inf.kyushu-u.ac.jp

シングサイトの運営，悪性プログラムの配布等，様々な脅威が存在する．これらの脅威のプラットフォームとなっているものがボットネットである．ボットネットは，ボットと呼ばれる悪性プログラムに感染した端末がネットワークを構築したものである．ボットは攻撃者によって遠隔操作され，攻撃者からの命令を受信することで，攻撃を開始する．

ボットネットは，分散サービス不能攻撃（DDoS，Distributed Denial of Service Attack）やフィッシングサイトの運営等，インターネット上の脅威のプラットフォームとして利用される [1], [2]．分散サービス不能攻撃では，ボットに感染した複数のホストが，攻撃者からの命令を受けて同時に攻撃を開始する．個々のボットは，少量のパケットを送信するだけで攻撃が成立するため，ホストの所有者に気付かれずに攻撃を実行することができる．また，ボットに感染した端末は，フィッシングサイトとしても利用される [3]．フィッシングサイトでは，ユーザの個人情報やパスワード，クレジットカード情報等が盗取される．攻撃者は，フィッシングサイトが発見されたとしても，別のボットをフィッシングサイトにすることによって，運用を継続することができる．さらに，ボットネットは，スパム送信のためのインフラとしても利用されている [4]．このようなボットネットを停止させることが課題であり，個々のボットに感染したホストをいち早く発見し，シャットダウンあるいは隔離する等の対処が必要となる．

1.2 問題提起

既存のボット検知手法として，シグネチャマッチングによる手法 [5], [6] や，ボットの協調性に着目した手法 [7], [8] が提案されている．シグネチャマッチング手法では，既知のボットの検知は可能でも，未知のボットについては検知漏れが発生してしまう．また，一方，ボットの協調性に着目した手法では，同じボットネットに所属しているボットが観測ネットワーク上に複数存在し，協調して動作する場合を仮定している．そのため，観測ネットワーク上にボットが 1 台しか存在しない場合は，検知が難しくなる．

ボットネットを停止させるには，それらのボットをとりまとめている中央サーバや，ボットネットの管理者を追跡することが効果的である．ボットネットの追跡を試みている研究もいくつかある．しかしながら，海外のプロキシを利用したり，何段もの踏み台を介したりして命令を送信するため，追跡は困難である．そのため，ボットに感染したホストを発見し駆除する手法を検討していかなければならない．

1.3 貢献

本研究では，機械的な通信と人間が行う通信の特徴の違いに着目したボット検知手法を提案する．ボットの挙動はプログラムによって規定されているため，機械的な挙動を持つ通信を発見することができれば，ボットを発見することができる．提案手法では，アプリケーションのネットワーク挙動から，そのアプリケーションが人間によって操作されているか，あるいはボットのようなプログラムによって動作しているかを特定することによって，機械的通信を発見する．

本論文の貢献は次のとおりである．

- はじめに，ネットワークアプリケーションの入力と出力の関係をモデル化を行う．また，人間が操作する IRC クライアントの IRC メッセージ送信間隔ならびにボットの IRC メッセージ送信間隔のモデル化を行い，そのモデルに従ったデータを作成する．作成したデータが，実際の IRC クライアントならびに IRC ボットの IRC メッセージ送信間隔分布に従っていることを示す．
- IRC メッセージ送信間隔のモデルの違いに着目したボット検知手法を提案する．モデルに従ったデータを用いて，人間的挙動と機械的挙動のモデルの違いを数値化する手法について検討し，人間か機械かを判定するアルゴリズムを設計する．
- モデルに即したデータを用いた評価ならびに実際のデータを用いた評価を行う．検知アルゴリズムの各パラメータについて，モデルに即したデータを用いながら考察を行う．そして，各パラメータの具体的な値を決め，実際の IRC クライアントのトラフィックならびに IRC ボットのトラフィックを用いて評価を行う．

1.4 本論文の構成

2 章では，関連研究について述べる．3 章では，アプリケーションの通信挙動のモデル化について検討する．4 章では，検討したモデルに従ったボット検知手法について提案する．5 章では，提案した検知アルゴリズムの評価を行う．6 章は結論を述べる．

2. 関連研究

人間と機械の違いに着目した研究として，Dews らは，ネットワーク上を流れるトラフィックから，インターネットチャットシステムによって生成されたトラフィックを発見する手法について提案している [9]．また，インターネットチャットにおける，セッションの時間間隔の分布や，パケットの到着時間の分布，チャットメッセージサイズの分布について調査を行っている．その中で，人間が操作するチャットシステムにおけるパケット到着時間の分布は，指数分布に類似したものになることを示している．この研究では，悪性ソフトウェアといったプログラムの挙動に関する記述はない．

Ma らは，人間のチャットメッセージのサイズと，ボット

のメッセージサイズの違いに着目したボット検知手法を提案している [10]．Ma らの結果では，ボットに感染したホストの TCP 通信におけるパケットサイズ列ならびにそのコンテンツ（Conversation Content Sequence，以降 CCS）が，周期性を持つことを示している．一方，人間の CCS はランダムに変化し，周期性を持たないことを示している．また，パケットサイズ自体も，人間の IRC チャットはボットの通信に比べてサイズが大きいことを示している．

Giavecchio らは，インターネットチャットシステムにおけるチャットボットの検知アプローチとして，人間のメッセージの送信間隔とメッセージの内容が，チャットボットのそれと異なることを用いて，チャットボットの検知を行っている [11]．チャットボットとは，インターネットチャットシステムにおいて，スパムや悪性 URL をクリックさせることを目的としたメッセージ送信自動化プログラムであり，本研究の対象としているボットとは直接的には関係しないが，人間と機械の挙動の違いに着目している点は類似している．Giavecchio らは，メッセージ送信間隔のエントロピーを定義し，人間の送信するメッセージのエントロピーとチャットボットのそれとが異なることを利用して，チャットボットの検知を行っている．

Akiyama らは，ボットネットを発見するための 3 つのメトリクスとして，ホスト間の関係，レスポンスタイム，同期に着目することが有効であることを示している [12]．また，本研究の先行研究として，Kugisaki らは人間の操作による IRC メッセージの送信間隔とボットによる IRC メッセージの送信間隔に違いがあることを示している [13]．両者とも，ボットに機械的な挙動が存在し，ボット検知に応用できることを示唆しているが，ボット検知アルゴリズムの設計とその性能評価については行っていない．

Gu らは，ボットに感染しているホストを発見する，BotMiner [8] や BotSniffer [7] を提案した．BotMiner は，ボットネットの構造に依存しないボット検知手法である．C&C サーバを通して命令を受信し協調的に動作するマルウェアのことをボットと定義し，類似した特徴を持つ通信トラフィックをクラスタリングし，さらにクラスタ間の類似度を計算することで，同じボットネットに所属しているボット群を発見することができるものである．BotSniffer は，ネットワークトラフィックに潜むボット・C&C サーバ間通信を検知するシステムである．シグネチャ等の情報をあらかじめ必要とせず，ネットワークに存在する複数のボットの通信が時間的あるいは空間的な協調性と類似性を持つことに着目している．Gu らは，ボットが各々同じ動きをするよう，あらかじめプログラムされたソフトウェアであるため，そのような類似性が現れることを主張している．BotSniffer も，BotMiner と同様，複数のボットが協調動作することに着目している手法である．これらの手法は，観測対象となるネットワークに，同じボットネットに所属する複数のボットが存在し，それらの通信を観測できることが条件となっている．そのため，観測対象に単一のボットしか存在していなかった場合は，協調動作を発見できないため，検知が難しい．対して，我々の手法は 1 つのボットと C&C サーバ間の通信が観測できればボットを検知することが可能となる．

3. 機械的挙動とアプリケーションプロトコルメッセージ送信間隔

3.1 ネットワークアプリケーションに対する入力と出力の関係

我々は，ネットワークアプリケーションの入出力とその挙動のモデルを作成した．**図 1** は，そのモデルを示している．アプリケーションに対する入力として，外部からのパケットならびにキーボード・マウスといったインタフェースからの入力がある．アプリケーションは，これらの入力に対して内部処理を行った後，外部に対して出力を行う．また，アプリケーションには，タイマ等の内部状態変化に従って処理を行う部分が存在し，ネットワークに対して出力を行う場合がある．

人間がアプリケーションを操作する場合は，アプリケーションから情報を受け取り，思考した後にキーボード操作等を経て，アプリケーションに対して入力を行う．そのため，人間の思考時間やインタフェースの操作時間が，アプリケーションへの入力に反映され，結果的にアプリケーションの出力にも影響を与える．思考時間や，インタフェースの操作時間は，人間が受け取る情報ごとに毎回変化するため，ランダム性を持つものになる．しかし，ボット等のプログラムの場合は，コードに記述されている挙動，すなわち，内部の状態変化による挙動に限定される．

3.2 アプリケーションの入出力観測と内部状態の推測

ネットワークアプリケーションがこのモデルに従うとすると，アプリケーションの出力を第三者が観測することによって，その入力の特徴や状態変化の様子を推測することができる．

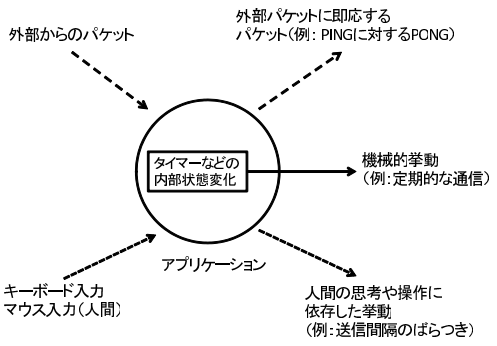

図 1 ネットワークアプリケーションの入出力モデル
Fig. 1 Input and output model for a network application.

たとえば，アプリケーション外部からの入力パケットに対して即座に応答する実装の場合，外部からの入力パケットを観測してすぐに，アプリケーションから送信されたパケットを観測することができる．そのような例として，IRC プロトコル [14] に従うメッセージ（本論文では「IRC メッセージ」とする）の一種である，PING/PONG メッセージがあげられる．PING/PONG メッセージは，サーバとクライアントの死活監視に用いられるメッセージであり，定期的にサーバからクライアントへ PING メッセージが送信される．IRC クライアントは，PING メッセージを受信すると，即座に PONG メッセージをサーバに対して送信する．

また，アプリケーション内部にタイマ等の実装が存在し，定期的にメッセージを出力する場合は，観測されるパケットの時間間隔は一定となる．一方，人間がキーボードを使って入力を行いメッセージ送信する場合，観測されるパケットの送信間隔は，キー入力の間隔に依存することとなり，その結果メッセージ送信間隔がばらついてくる．

ネットワークアプリケーションのパケット出力から，そのアプリケーションを操作している主体が人間か機械かを判定する関連研究として，Xie らのウェブブラウジング挙動を基にした異常検知手法 [15] がある．Xie らの研究では，ウェブブラウザからの HTTP リクエストメッセージに対して，隠れセミマルコフモデルを適用し，ブラウザを操作している主体が人間かそうでないかを判別することで，ウェブサーバに対する攻撃判定を行っている．この研究では，観測した HTTP リクエストメッセージのみを用いて，ウェブブラウザの機械的挙動の判別を試みている．

3.3 IRC クライアントにおける具体例

ネットワークアプリケーションの具体例として，IRC クライアントに着目する．IRC クライアントの場合，それが人間によって操作されているクライアントか，あるいはボットのようなプログラムであるかによって，観測される通信挙動に違いが現れる．IRC クライアントを人間が操作している場合，同じような挙動を繰り返すことや，同じタイミングで IRC メッセージを送信することは稀になる．たとえば，チャネル内でニックネームを変えるときに NICK コマンドが発行されるが，人間の場合は NICK コマンドが成功すれば，それ以上は NICK コマンドを発行しなくなる．また，チャットの相手とチャットメッセージのやりとりをする場合，チャットメッセージを受け取ってから思考し，キーボードで文字を入力して送信するまでの時間は毎回変化し，IRC メッセージの送信間隔はばらついたものとなる．一方，IRC ボットの場合，次にあげる状況で IRC メッセージが定期的に送信される [10]．

(a) C&C サーバが稼働しており，IRC ボットが TCP コネクションを維持するために，C&C サーバから送信された PING メッセージに対して，PONG メッセージを応答する挙動を繰り返すような状況

(b) C&C サーバに接続できない場合であっても，IRC ボットがハードコードされている C&C サーバへの接続を定期的に試みる状況

(c) ボットネットがプログラムによって管理されており，そのプログラムが定期的にボットに対して命令を送信するような状況

これらは，ボットないしボット管理者の挙動がコードによって記述されており，コードに従った挙動しかとることができないために生じる現象である．

3.4 IRC クライアントの IRC メッセージ送信間隔のモデル化

我々は，IRC クライアントが送信する IRC メッセージの送信間隔分布モデルを作成した．このとき，IRC メッセージとは，人間あるいはボットが入力したチャットのメッセージ，ならびに NICK や JOIN メッセージ等の IRC プロトコルにおける制御メッセージを指す．

人間が入力するチャットメッセージの送信間隔分布は，Dews らの結果に基づくと，指数分布によって近似できる．実際の IRC クライアントでは，チャットメッセージのほかに，定期的に送信される PING メッセージに対する PONG メッセージ等も観測されるため，厳密な指数分布ではない．しかし，今回のモデル化では，PONG メッセージ等も含めて，単純な指数分布であると見なす．

ボットがたかだか数種類の機械的挙動しかとらないと仮定すると，メッセージの送信間隔が，t_i（$i = 1, 2, 3, \ldots$）付近に集中する．このとき，i の値は，ボットがとりうる挙動の数に依存する．たとえば，ボットがログイン試行，スキャン命令の実行，スキャン結果の報告の 3 通りの挙動しかとらないとすると，$i = 1, 2, 3$ となる．メッセージの送信間隔は，t_i を中心に $\pm\Delta t$ 以内に集中しており，その部分に全体の α のメッセージが集中しているとする．**図 2** の例では，i が 3 のとき，t_1，t_2，t_3 それぞれの部分に α_1，α_2，α_3 のメッセージが集中している．このとき，$T \leq \alpha_1 + \alpha_2 + \alpha_3 \leq 1$ とする．T は，機械的な挙動であることを示すため，ある程度大きい値とする．

3.5 提案モデルに基づく IRC クライアントの IRC メッセージ送信間隔分布

前節で提案したメッセージ送信間隔のモデルについて，擬似的なデータを生成し，その分布について確認した．人間が操作した IRC クライアントの IRC メッセージ送信間隔の分布は，指数分布で近似できる．指数分布に基づく乱

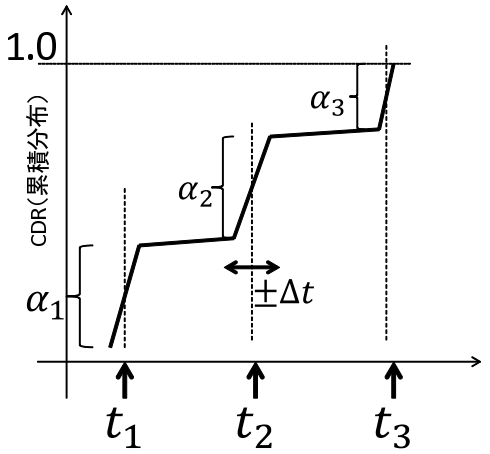

図 2 機械的挙動における IRC メッセージ送信間隔の分布モデル
Fig. 2 Distribution model of IRC message transmission intervals for mechanical behaviors.

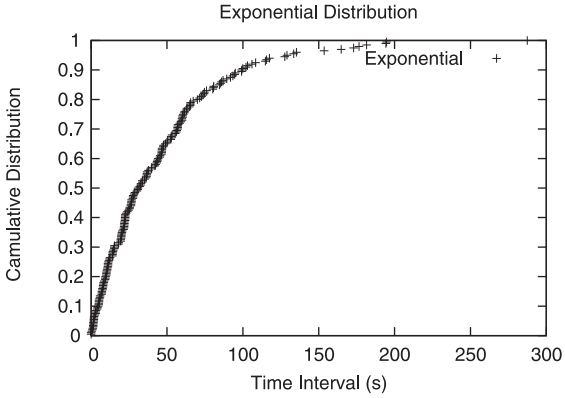

図 3 指数分布に基づく乱数の累積分布（データ数：200）
Fig. 3 Random value based on exponential distribution (# of data: 200).

数は，パラメータとして平均を決めることで生成できる．今回は，平均を 45 秒として計算した．この値は，4 人のテスタに IRC チャットを 1 時間行ってもらった際に得られた IRC メッセージのうち，チャットメッセージを含むものの送信間隔の平均である．**図 3** は，生成した乱数 200 個の累積分布を示している．

図 4 は我々が提案する機械的挙動モデルに基づき作成した，IRC メッセージ送信間隔を表すデータである．この図では，分布が集中している点を，$t_1 = 0$, $t_2 = 90$, $t_3 = 120$（単位：秒）とし，集中の度合いをそれぞれ $\alpha_1 = 0.2$, $\alpha_2 = 0.5$, $\alpha_3 = 0.1$ と設定した．$\Delta t = 0.001$ とし，0.001 以下で生成した乱数を t_k に付加して揺らぎを表現した．図に描かれているデータ数は 200 個である．

3.6 実データにおける IRC メッセージ送信間隔の累積分布

提案モデルに従って生成したデータと，現実のデータ比較のため，人間が操作する IRC クライアントのトラフィック，ならびに IRC ボットの通信トラフィックについて，

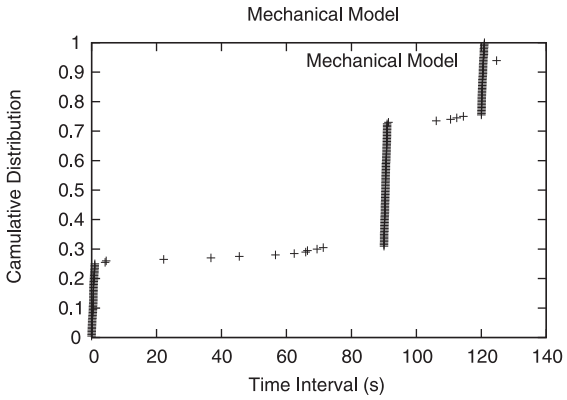

図 4 機械的モデルに基づく擬似データ（データ数：200）
Fig. 4 Artificial data based on mechanical behavior model (# of data: 200).

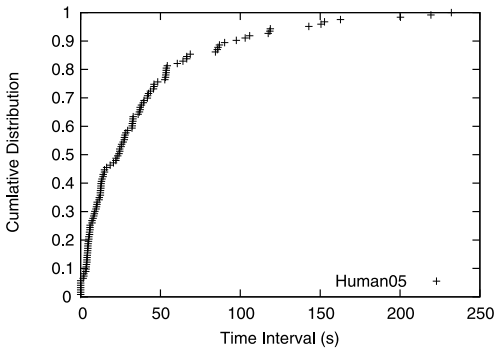

図 5 人間が操作する IRC クライアントのチャットメッセージ送信間隔分布（その 1，データ数：120）
Fig. 5 Distribution of message transmission intervals for human-operated IRC client (No.1, # of data: 120).

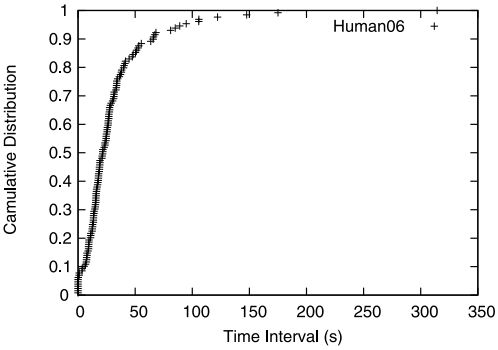

図 6 人間が操作する IRC クライアントのチャットメッセージ送信間隔分布（その 2，データ数：130）
Fig. 6 Distribution of message transmission intervals for human-operated IRC client (No.1, # of data: 130).

その IRC メッセージ送信間隔の累積分布を描いた（**図 5**, **図 6**, **図 7**）．人間の IRC トラフィックは，大学ネットワーク内でテスタにチャットを行ってもらい，その IRC トラフィックを取得した．また，IRC ボットのトラフィックは，大学ネットワーク内で観測されたボットの通信を取得した．

図 5 は，大学ネットワーク内でテスターが行ったチャッ

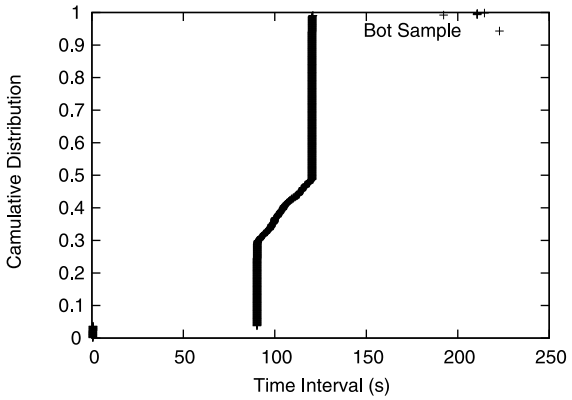

図 7 IRC ボットにおける IRC メッセージ送信間隔分布（データ数：816）

Fig. 7 Distribution of message transmission intervals for IRC bot (# of data: 816).

トの IRC トラフィックについて，PONG メッセージを取り除いた後，IRC メッセージ送信間隔の累積分布を描いたものである．IRC メッセージ送信間隔を示すデータ数は 120 個である．今回，PONG メッセージを除外した理由は，人間のチャットメッセージの送信間隔分布が，指数分布に近い分布をとることを示すためである．この図では，指数分布に類似したばらつきのある時間間隔で IRC メッセージが送信されていることが分かる．同じく図 6 は，他のテスタが行った IRC チャットのトラフィックについて IRC メッセージ送信間隔の累積分布を描いたものである．IRC メッセージ送信間隔を示すデータ数は 130 個である．この例も図 5 と同様に，指数分布に類似したばらついた分布を描いていることが分かる．ここで，送信間隔の平均の違いは個人差として表現することができる．具体的には，指数分布における平均の違いが，個人差にあたるものとなる．今回，テスタの平均の IRC メッセージ送信間隔が 45 秒となり，擬似データとして生成する指数分布に基づく乱数もこの値を利用している．

図 7 は，ボットに感染したホストと IRC サーバ間の IRC トラフィックについて，データ送信間隔の分布を描いたものである．IRC メッセージ送信間隔を示すデータ数は 816 個である．人間の場合に比べて，その分布の偏りがはっきりと現れている．この IRC トラフィックでは PONG メッセージは観測されず，ボットが定期的に NICK コマンドを発行している挙動が観測された．これは，ボットが IRC メッセージを送信する理由の (b) に該当する．

実験結果を見ると，ボットの通信の場合，$t_1 = 90$ 秒付近と $t_2 = 120$ 秒付近に送信間隔が集中していることが分かる．$t_1 = 90$ 秒から前後 $\Delta t = 1$ 秒以内に，$\alpha = 0.263$ のプロットがあり，$t_2 = 120$ 秒から前後 $\Delta t = 1$ 秒以内に，$\beta = 0.512$ のプロットが存在する．一方，人間が操作する IRC クライアントの IRC トラフィックの場合，そのように送信間隔が集中している部分は見当たらない．これにより，人間が生成した IRC トラフィックの IRC メッセージ送信間隔の分布と，ボットが生成した IRC メッセージ送信間隔の分布には違いが確認された．

4. 検知アルゴリズム

4.1 基本的なアイデア

続いて我々は，IRC メッセージの送信間隔から，IRC メッセージが人間の操作によって出力されているか，あるいは，IRC ボットのようなプログラムによって出力されているのかを判定するアルゴリズムについて検討する．前章で明らかにした，人間の IRC メッセージ送信間隔の分布と IRC ボットのそれとの違いを数学的に表現し，人間の通信とボットの通信を分離する手法について検討する．我々は，機械的な挙動では，IRC メッセージ送信間隔が特定の値に集中することに着目し，クラスタリングを用いた手法について検討する．我々は，IRC メッセージ送信間隔の列に対しクラスタリングを適用すると，送信間隔の分布の特徴を抽出できるのではないかと考えた．

4.2 クラスタリング

クラスタリング手法には，大きく分けて，階層型クラスタリングと非階層型クラスタリングが存在する．階層型クラスタリングの代表例は凝集型クラスタリングで，重心法や群平均法，ウォード法等がある．凝集型クラスタリングでは，初期状態では，入力データの各々を 1 つのクラスタと見なし，そのクラスタ間距離が最小となるものを順番に統合していく．最終的には，すべての要素が 1 つのクラスタに集約される．凝集型では，すべてのクラスタ間距離を計算するため，N 個の入力に対し (N^2) の計算量がかかるが，結果は 1 通りとなる．距離の計算方法として，重心法や群平均法等が存在するが，今回は重心法を用いる．クラスタリングには，統計処理ソフトウェア R [16] を用いた．R では，階層型クラスタリング処理のための関数が用意されており，重心法が指定できる．距離の計算はユークリッド距離を用いている．

凝集型階層的アルゴリズムを以下に示す．

(1) 開始時は，クラスタリングの対象となる各要素（ここでは数値）を 1 つのクラスタと見なし $(C = (c_1, c_2, \ldots, c_n))$，アルゴリズムの入力とする．
(2) 同時に，各クラスタ間の類似度を示す類似度行列を作成する．開始時は単純な数値の差を類似度と見なす．
(3) 類似度行列から距離が最小となる 2 つのクラスタを選択し，集約する．この際，新しいクラスタの重みを計算する必要があるが，これを重心法によって計算する．
(4) 類似度行列を更新する．
(5) (2)〜(4) を繰り返す．

非階層型クラスタリングの例は k-means 法である．k-means 法では，初期状態を任意に決定した後，局所解を求

める手法であり，設定した初期状態によって結果に大きく影響するため，初期状態をランダムに変更したものを k 回繰り返し，評価関数を最も小さくするものを結果として選ぶ．このように，繰返しが必要で効率が悪いため，今回は利用しない．

4.3 挙動の数値化

はじめに，人の操作による IRC メッセージ送信間隔モデル，ならびに機械による IRC メッセージ送信間隔モデルに対して階層型クラスタリングを適用し，要素の集約されかたの違いを確認した．それぞれのデータとして，3.4 節で生成したデータを用いている．階層型クラスタリングは重心法を用い，入力として，モデルに基づいて生成した数値列を各 200 個用いた．集約の様子は，クラスタリングが進行する途中の，各クラスタに含まれる要素数をカウントすることによって確認した．

図 8，**図 9** は，3.4 節で作成した，人間の IRC メッセージ送信間隔モデルに従ったデータならびに，機械の IRC メッセージ送信間隔モデルに基づくデータをクラスタリングした結果である．横軸は，クラスタ数を示しており，値が小さくなるほどクラスタリングが進行していることを示している．縦軸は，ある時点 k における各クラスタに含まれる要素数の割合を示している．図 8，図 9 では，クラスタの統合が 10 回行われるごとに要素数のカウントを行っている．グラフは，要素数の多いものから 1 位～10 位，その他の順に並べられている．

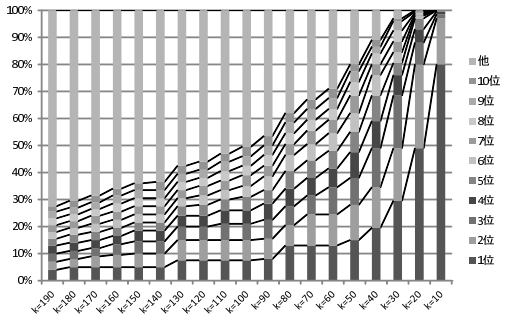

図 8 指数分布に基づく擬似データに対するクラスタリング結果
Fig. 8 Clustering against artificial data based on exponential distribution.

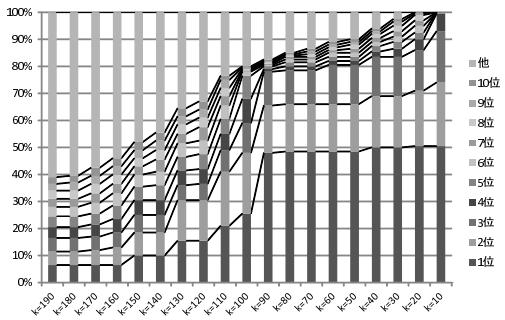

図 9 機械的通信挙動モデルに基づく擬似データに対するクラスタリング結果
Fig. 9 Clustering against artificial data based on mechanical communication behavior model.

図 8 では，各クラスタに含まれる要素数がゆるやかに上昇していく様子が確認できる．これは，人間の IRC メッセージ送信間隔モデル，つまり指数分布がばらついているためで，各クラスタに含まれる要素数の割合は，異なる k の値においても同じくらいの割合となっている．図 8 では，$k = 40$ 付近までは緩やかに集約が進み，$k = 40$ 以降，急速に集約が進んでいる．

一方，図 9 では，$k = 100$ 付近から，上位 3 つのクラスタに含まれる要素数の割合が増加しており，$k = 90$ 以降，その 3 つのクラスタに含まれる要素数の割合は安定している．これは，3.4 節で提案した機械的挙動に基づくモデルによる IRC メッセージ送出間隔分布が 3 つのパラメータ t_1，t_2，t_3 の周辺に集中するためであり，$50 < k < 90$ の範囲で，それらの要素は要素数の順に上位 3 つのクラスタに集中するためである．一方，人間が操作する場合の指数分布を用いたモデルによる IRC メッセージ送出間隔分布は，特定の値に集中することがないため，上位 3 つのクラスタに属する要素が 0.8 を超えるのは $k < 30$ の範囲となる．

クラスタリングの結果，人間のモデルの場合は，要素が徐々に集約されているのが見えるが，ボットの場合は，$k = 50$ において，最も多い 3 つのクラスタが約 80%を占めていることが分かる．そこで，ある時点での上位いくつかのクラスタに含まれる要素数が，人間と機械で違うことを利用して，機械的かどうかの判定を行う．具体的には，ある時点での上位のクラスタに含まれる要素数の割合に，指数分布と対象データとの間で一定以上の差がある場合，対象データは機械的なモデルに従っていると判断する．

4.4 機械的挙動の検知アルゴリズム

クラスタリングの結果から，以下のアルゴリズムを設計した．

(1) IRC メッセージ送信間隔の列 $(t_1, t_2, t_3, \cdots, t_N)$ を階層型クラスタリングの入力とする．N は IRC メッセージ送信間隔の総数とする．
(2) IRC メッセージ送信間隔列に対して階層型クラスタリングを適用する．集約が 1 回行われるごとに，上位 M 個のクラスタに含まれる要素数の割合を計算し，比較対象となる指数分布に基づく擬似データにおける要素数の割合との差をとる．
(3) 差が最大になった時点で人間対機械の判定を行う．差が閾値 T 以下になる場合は人間と判定する．閾値 T 以上になる場合は機械と判定する．

ここで，3 つのパラメータ N，M，T ならびに，割合の差を最大にする点に関する議論が必要となる．パラメータの設定については，5 章で考察を行う．

5. 評価

本章では，我々が提案する機械的挙動の検知アルゴリズムにおけるパラメータ設定に関する議論，ならびに実際のIRCトラフィックを用いた提案手法の評価を行う．

5.1 各パラメータに関する議論

はじめに，検知アルゴリズムにおける3つのパラメータ（N，M，T）ならびに，判定を行うタイミングについて考察を行う．それぞれのパラメータで検討すべき事項は，次のとおりである．IRCメッセージの総数Nは，どれくらいの数のIRCメッセージを観測すれば，人間の通信と機械的通信を区別可能であるかにかかわる．比較するクラスタ数Mは，どれくらいのクラスタを選択すれば人間と機械を区別可能であるかにかかわる．これは，機械のIRCメッセージ送信間隔の分布モデルにおける，t_kのkの数に依存する．判定基準となる閾値Tは，人間と機械を区別するのに十分な値に設定する必要がある．最後に，判定を行うタイミングについて議論する．それぞれ，3.4節で示した，人間と機械の典型的なIRCメッセージ送信間隔分布モデルを用いて考察を行う．

5.1.1 比較対象となるクラスタ数 M について

要素数の比較を行う際に選択するクラスタの数Mの値は，機械的分布モデルに現れるブロックの数に依存する．3.4節のモデルに従う場合，t_1，t_2，t_3に値が集中するため，$M = 3$となる．ボットが，効率を重視して設計され，複雑な挙動をとらないと仮定すると，その挙動の種類はたかだか数個になり，結果的に集中する箇所も数カ所になる．集中する点が2カ所の場合は，提案するモデルにおいて$t_1 = t_2$とすれば同様の議論ができる．今回は，$M = 3$として議論をすすめる．

5.1.2 N について

次に，Nがどのような値の場合，本アルゴリズムが利用可能かを示す．$N = 50, 100, 200$の場合で擬似データを生成し，区別ができるかを考察した．上位3つ（$M = 3$）のクラスタに含まれる要素数の割合を，人間のIRCメッセージ送信間隔分布モデルと機械におけるモデルで比較し，区別可能であるかを目視で確認した．**図10**，**図11**，**図12**は，各Nの値におけるクラスタリング結果を示す．$N = 50$では，$k = 16$付近で差が最大となっているが，人間のモデルと機械のモデルを明確に区別可能な差ではない．統計的な観点から，$N = 50$では，指数分布モデルと機械的分布モデルに大きな差が現れず，誤検知が発生する可能性があるため，$N = 50$で利用することは避けたほうがよい．$N = 100$の場合では，$k = 26$付近に，$N = 200$では$k = 47$付近に最大の差が現れ，区別可能な値となっている．結論として，$N = 100$以上で本アルゴリズムを利用することが望ましい．

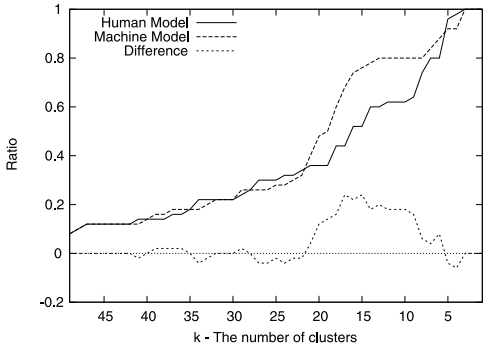

図10 $N = 50$のときの上位3つのクラスタに含まれる要素数比較
Fig. 10 Comparison between two artificial data ($N = 50$).

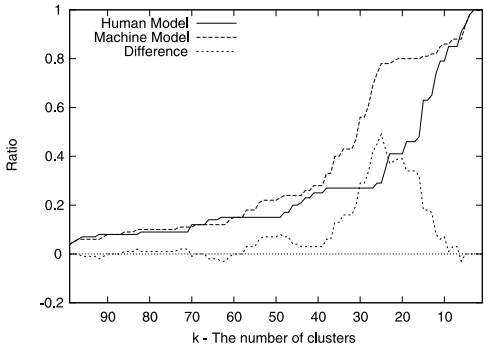

図11 $N = 100$のときの上位3つのクラスタに含まれる要素数比較
Fig. 11 Comparison between two artificial data ($N = 100$).

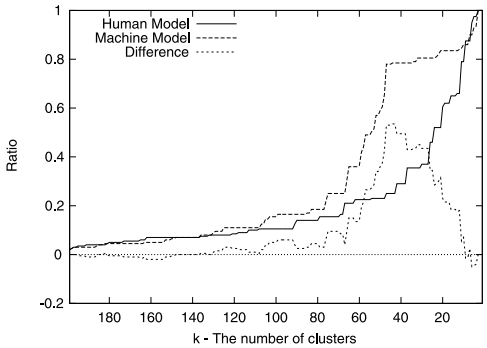

図12 $N = 200$のとき上位3つのクラスタに含まれる要素数比較
Fig. 12 Comparison between two artificial data ($N = 200$).

5.1.3 判定を行うタイミングについて

人間対機械の判定を行うタイミングは，各々の上位M個に含まれる要素数の差が最大となる点をとる．図12では，$k = 47$のときに，最大となる0.565の差が生じる．機械的モデルでは，$k = 50$付近から，上位3つのクラスタに含まれる要素数がほぼ一定となり，$k = 47$以降，差が小さくなるため，この時点を判定のタイミングとする．

5.1.4 T について

差が最大になる点において，人間対機械の判定を行う．判定対象となるデータが指数分布に近い値をとるとき，判定対象は人間であるとし，差が閾値T以上になる場合は，機械と判定する．閾値Tについては，図12のモデルを用いて決定する．図12では，機械モデルの要素数の割合が

急上昇する $k = 60$ 付近から，機械モデルの要素数が安定し，指数分布の要素数が急上昇する $k = 20$ 付近までに，差が 0.3 以上を維持しており安定している．そこで今回，機械判定を行う閾値を $T = 0.3$ とする．

5.2 実際の IRC トラフィックを用いた評価

5.2.1 データの準備

人間の IRC チャットのトラフィックは，大学内のテスタに協力してもらい，チャットのトラフィックを IRC サーバ上で取得した．トラフィックは 30 サンプル取得した．IRC クライアントとして，LimeChat [17] を用い，IRC サーバとして，CentOS 上にインストールした ngIRCd [18] を用いた．チャットは自由な話題で 1 時間から 2 時間かけて行い，各クライアントが送信したパケットを，IRC サーバ上で tcpdump により取得した．各サンプルに含まれるメッセージ数は，最小で 78 個，最大で 572 個観測された．

IRC ボットのトラフィックは，大学ネットワーク内で発見された IRC ボットのトラフィックが 5 サンプル，ならびにハニーポット運用によって取得された IRC ボットのトラフィックが 30 サンプルの，計 35 サンプルを用いた．大学ネットワークで取得されたサンプルは，2006 年に発見された 3 種類のボットの通信，ならびに 2011 年に発見された 2 種類のボットの通信を用いた．ハニーポットは 2010 年から 2011 年にかけて運用されたものを用いた．各サンプルに含まれるメッセージ数は，最小で 71 個，最大で 7,272 個観測された．

各サンプルは，IRC クライアントから IRC サーバに向けて送信されたパケットのうち，チャットメッセージならびに IRC メッセージが含まれているものを取り出して作成した．つまり，ACK のみのパケット等，メッセージが含まれていないパケットは除外している．またメッセージの個数は，先頭から 200 メッセージまでを抽出して，アルゴリズムに入力している．

5.2.2 提案手法における評価結果

図 13 にボット（IRC Bot）および人間（Human）が生成したトラフィックの評価結果を示す．図の横軸（Sample ID）はサンプルの ID を表しており，1 から 35 までを示している．図の縦軸（Diff）は，アルゴリズムによって計算された，基準となる指数分布との数値の差が最大となったものを示している．また，サンプル ID は数値の差が昇順となるようにソートした上で割り当てている．

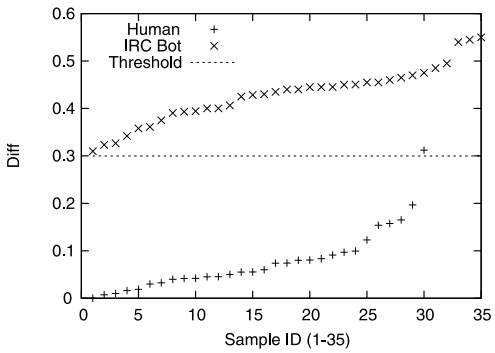

図 13 実験結果
Fig. 13 Evaluation result.

IRC ボットの 35 サンプルは，指数分布との差が 0.31 (B-01)～0.55 (B-35) となり，閾値 $T = 0.3$ の下ですべて「機械」と判定された．人間の IRC トラフィック 30 サンプルでは，H-01 から H-29 までの 29 サンプルにおいて，差が 0.00～0.20 となり，「人間」と判定された．一方，H-30 は差が 0.31 となり「機械」と判定された．

誤判定された人間のサンプル ID H-30 について分析を行った．人間のサンプル ID H-30 では，PONG メッセージの占める割合が，全体の 34%を占めていた．具体的には，全体で 125 個のメッセージのうち，PONG メッセージが 43 個存在した．このように，PONG メッセージが多くなる状態は，IRC サーバに接続したままチャットを行わない状態が続いたときであり，3.2 節に示したボットの (a) に該当する挙動に類似する．このように，PONG メッセージが多く存在すると誤検知が発生する．この点については，Ma らの手法においても，チャットメッセージが極端に少ない状況において，同様の誤検知が発生することが言及されている．本手法は，ボット検知の観点から，見逃し率を小さくすることを優先する．誤検知は，他の手法を併用するシステムを構築することによって回避しうると考えている．

5.2.3 既存手法における評価結果

続いて，同一の評価実験サンプル（ボット 35 サンプルおよび人間 30 サンプル）を用いて，既存手法の 1 つである Gianvecchio らの手法 EN-imd [11] を用いて評価実験を行った．EN-imd では，インターネットチャットに流れる機械生成によるメッセージを，そのメッセージの送信間隔に現れる特徴に基いて検知することを試みている．我々の提案する手法と同様に，メッセージの送信間隔に着目している手法として，提案手法の比較対象とした．

EN-imd では，検知システムを動作させる前段階として，いくつかパラメータの設定が必要となる．今回はエントロピー計算のためのパラメータとして，事象区間サイズを 1.0 秒とし，履歴参照は行わないとした．事象区間サイズ 1.0 秒という値は，3.6 節において図 7 に示した IRC ボットの送信間隔の集中を示すパラメータ $\Delta t = 1$ 秒以内を基にした．**図 14** は，EM-imd で用いているエントロピー計算手法を基に，サンプルのエントロピーを計算した結果を示している．

図 14 の横軸はサンプル ID を示しており，縦軸は EN-imd によって計算されたエントロピーの値を示している．サンプル ID はエントロピーの値が昇順になるようにソートしたうえで割り当てている．EN-imd では，事前に観測した，人間の操作による挙動のエントロピーの分布を基に，

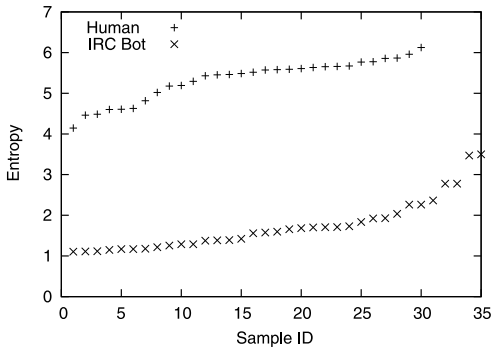

図 14 実験結果 EN-imd
Fig. 14 Evaluation result (EN-imd).

ボット判定のための閾値を決定している．参考文献では，事前学習のサンプルの 99%を含むところに閾値を設定しているため，人間の 30 個のサンプルのうち任意の N 個を用いて判定閾値を決定した場合，見逃しは起こらなかった．一方，学習するサンプルによっては誤検知が生じる結果となった．たとえば，最悪のケースとして，事前学習においてサンプル ID10 から 30 までが選択され閾値が決定された場合，少なくとも ID01 から 07 までは，誤検知となる．

5.2.4 提案手法と既存手法の比較および考察

続いて，既存手法との比較を行う．サンプルデータセットによる評価では，Gianvecchio らの手法の EN-imd との比較を行った．関連研究中の EN-imd は，人間の操作によるメッセージ送信間隔の特徴を観測パケットにより事前学習し，判定の閾値を決定している．一方，我々の提案手法は，人間のメッセージ送信間隔分布を，指数分布による数理的モデルで表現している．この数理的モデルのパラメータは，人間の操作によるメッセージ送信間隔の平均のみから決定できるため，事前学習が先行研究 EN-imd の場合より単純となる．また指数分布は，その平均が変化しても，それにより生成される値をクラスタリングした結果は大きく変化しないため，ネットワーク遅延によりメッセージ送信間隔の平均が変化する場合でも，検知率の低下は抑えられる．

ボット検知性能に関して，見逃しについては，提案手法でも既存手法でもゼロとなり，同等の結果となった．ボット検知における誤検知は，提案手法では 1 例にとどまった．一方，既存手法では，学習するサンプルの分布によっては誤検知が生じることとなった．また，既存手法はエントロピー算出にあたり適切な事象区間を決定するためのパラメータを適切に設定する必要があることからも，我々の提案手法は既存手法と比較して単純であり実用的といえる．

Ma らの評価実験では，ランダムにコマンドを選び IRC サーバに送信するボットが，「人間」として判定されている．そのため，メッセージの送信間隔だけに着目している我々の手法では，そのようなボットを検知できると考えられる．

次に，我々の手法によるボット検知を回避する手法について考える．我々の提案手法に対する対策として，ボットが IRC メッセージの送信間隔を変化させる手法が考えられる．たとえば，指数分布に基づく乱数時間待機してメッセージを送信するような実装にすると，「人間」として判定される．そのような手法に対しては，人間と判断される範囲に下限を設定し，送信間隔の分布が数学的モデルに極端に近い場合は「機械」と判定するといった対策が検討できる．

本手法では，N の値が小さくなるにつれて，データに含まれる階層的な構造が見えにくくなるため，N が小さいと性能が低下する可能性がある．十分な性能を得るためには，$N = 100$ 以上が望ましく，取得できるパケットがそれ以下の場合は適用が難しくなる．これについては今後の課題とする．

最後に，本手法は，IRC クライアントが機械的な挙動をとっているか，そうでないかを判定するものであり，機械的な挙動をとっている IRC クライアントが本当にボットであるかどうかまでは判断しない．しかしながら，大量のトラフィックの中から機械的な挙動を持つ通信だけを抽出することは，ボットの通信を発見するための初期段階として効果的である．

6. 結論と今後の課題

本論文では，IRC メッセージの送信間隔の観測により IRC ボットの通信を発見する手法として，人間と機械の通信挙動の違いに着目したボット検知手法を提案した．はじめに，ネットワークアプリケーションの入出力モデルならびにそのアプリケーションプロトコルメッセージ送信間隔のモデルを作成し，人間が操作するアプリケーションの通信挙動と，ボットの通信挙動の違いについて述べた．さらに，この違いを階層型クラスタリングによって数理的に表現する手法について検討し，それを用いた人間対機械判定アルゴリズムを設計した．評価では，検知アルゴリズムのパラメータ設定について議論した後，人間が操作する IRC クライアントの IRC トラフィックと IRC ボットのトラフィックを用い，検知性能の評価を行った．その結果，IRC ボット 35 サンプルがすべて機械として判定された．

今後の課題として，IRC ボットに限らないその他のプロトコル，たとえば HTTP ベースのボットに対して，本手法適用の検討を行っていく．HTTP ベースのボットの場合，ボットは定期的に司令サーバにアクセスする必要があるため，機械的な特徴は出やすいと考えられる．一方，HTTP は正規のプログラムも多く利用するプロトコルであり，誤検知の問題が発生する．その点の改善を今後の課題とする．

謝辞 本研究は，「国際連携によるサイバー攻撃の予知技術の研究開発（総務省）」の支援を受けている．

参考文献

[1] Freiling, F., Holz, T. and Wicherski, G.: Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks, *Proc. 10th European Symposium on Research in Computer Security, ESORICS*, pp.319–335 (2005).

[2] Nazario, J. and Holz, T.: As the net churns: Fast-flux botnet observations, *Proc. 3rd International Conference onMalicious and Unwanted Software 2008* (*MALWARE 2008*), pp.24–31, IEEE (2008).

[3] Stone-Gross, B., Cova, M., Cavallaro, L., Gilbert, B., Szydlowski, M., Kemmerer, R., Kruegel, C. and Vigna, G.: Your botnet is my botnet: Analysis of a botnet takeover, *Proc. 16th ACM Conference on Computer and Communications Security*, pp.635–647, ACM (2009).

[4] Pathak, A., Qian, F., Hu, Y., Mao, Z. and Ranjan, S.: Botnet spam campaigns can be long lasting: Evidence, implications, and analysis, *Proc. 11th International Joint Conference on Measurement and Modeling of Computer Systems*, pp.13–24, ACM (2009).

[5] Sourcefire, Inc.: SNORT, Sourcefire, Inc. (online), available from ⟨http://www.snort.org/⟩ (accessed 2012-12-20).

[6] Goebel, J. and Holz, T.: Rishi: Identify bot contaminated hosts by irc nickname evaluation, *Proc. 1st Conference on First Workshop on Hot Topics in Understanding Botnets* (*HotBots'07*) (2007).

[7] Gu, G., Zhang, J. and Lee, W.: BotSniffer: Detecting botnet command and control channels in network traffic, *Proc. 15th Annual Network and Distributed System Security Symposium* (*NDSS'08*) (2008).

[8] Gu, G., Perdisci, R., Zhang, J. and Lee, W.: BotMiner: Clustering analysis of network traffic for protocol-and structure-independent botnet detection, *Proc. 17th Conference on Security Symposium*, pp.139–154, USENIX Association (2008).

[9] Dewes, C., Wichmann, A. and Feldmann, A.: An analysis of Internet chat systems, *Proc. 3rd ACM SIGCOMM Conference on Internet Measurement*, pp.51–64, ACM (2003).

[10] Ma, X., Guan, X., Tao, J., Zheng, Q., Guo, Y., Liu, L. and Zhao, S.: A Novel IRC Botnet Detection Method Based on Packet Size Sequence, *Proc. 2010 IEEE International Conference on Communications* (*ICC*), pp.1–5, IEEE (2010).

[11] Gianvecchio, S., Xie, M., Wu, Z. and Wang, H.: Humans and Bots in Internet Chat: Measurement, Analysis, and Automated Classification, *IEEE/ACM Trans. Networking*, Vol.19, No.5, pp.1557–1571 (2011).

[12] Akiyama, M., Kawamoto, T., Shimamura, M., Yokoyama, T., Kadobayashi, Y. and Yamaguchi, S.: A proposal of metrics for botnet detection based on its cooperative behavior, *Proc. International Symposium on Applications and the Internet Workshops 2007* (*SAINT Workshops 2007*), pp.82–82, IEEE (2007).

[13] Kugisaki, Y., Kasahara, Y., Hori, Y. and Sakurai, K.: Bot detection based on traffic analysis, *Proc. 2007 International Conference on Intelligent Pervasive Computing* (*IPC2007*), pp.303–306, IEEE (2007).

[14] Oikarinen, J.: RFC1459, Internet Relay Chat Protocol (IRC), The Internet Engineering Task Force (IETF) (online), available from ⟨http://tools.ietf.org/html/rfc1459.html⟩ (accessed 2012-12-20).

[15] Xie, Y. and Yu, S.-Z.: A large-scale hidden semi-Markov model for anomaly detection on user browsing behaviors, *IEEE/ACM Trans. Networking*, Vol.17, No.1, pp.54–65 (2009).

[16] Gentleman, R. and Ihaka, R.: The R Project for Statistical Computing, R Project (online), available from ⟨http://www.r-project.org/⟩ (accessed 2012-12-20).

[17] Nakagawa, S.: LimeChat, LimeChat (online), available from ⟨http://limechat.net/⟩ (accessed 2012-12-20).

[18] Barton, A.: ngIRCd: Next Generation IRC Daemon, ngIRCd (online), available from ⟨http://ngircd.barton.de/⟩ (accessed 2012-12-20).

溝口 誠一郎

昭和 60 年生．平成 20 年九州大学工学部電気情報工学科卒業．平成 22 年九州大学大学院システム情報科学府情報工学専攻修士課程修了．同年九州大学大学院情報学専攻博士課程進学．電子情報通信学会会員．

笠原 義晃 （正会員）

昭和 44 年生．平成 3 年九州大学情報工学部情報工学科卒業．平成 5 年九州大学大学院工学研究科情報工学専攻修士課程修了．平成 8 年九州大学大学院工学研究科情報工学専攻博士後期課程修了．博士（工学）．平成 8 年より九州大学大型計算機センター助手．平成 19 年より改組により九州大学情報基盤研究開発センター助教．インターネット管理運用，学内外の学術ネットワークに関する研究，情報セキュリティに関する研究に従事．電子情報通信学会会員．

堀 良彰 （正会員）

昭和 44 年生．平成 4 年九州工業大学情報工学部電子情報工学科卒業．平成 6 年九州工業大学大学院情報工学研究科情報システム専攻修士課程修了．平成 6 年九州芸術工科大学助手．博士（情報工学）．平成 16 年より九州大学大学院システム情報科学研究院助教授（現，准教授）．平成 17 年より 18 年にかけてカリフォルニア大学アーバイン校計算機科学部訪問研究員．情報ネットワーク，ネットワークセキュリティ，コンピュータシステムセキュリティ等の研究に従事．電子情報通信学会，IEEE，ACM 各会員．

櫻井 幸一 （正会員）

昭和 38 年生．昭和 61 年九州大学理学部数学科卒業．昭和 63 年九州大学大学院工学研究科応用物理専攻修士課程修了．昭和 63 年三菱電機株式会社入社後，情報電子研究所（現，情報総合研究所）にて，暗号と情報セキュリティの研究開発に従事．博士（工学）．平成 6 年より九州大学工学部情報工学科助教授．平成 9 年より 10 年にかけて米国コロンビア大学計算機科学科訪問研究員．現在，九州大学大学院システム情報科学研究院教授．平成 16 年より財団法人九州システム情報技術研究所第 2 研究室長（現，財団法人九州先端科学技術研究所情報セキュリティ研究室長）．電子情報通信学会，日本数学会，ACM，IEEE 各会員．