情報処理学会論文誌 Vol.56 No.9 1745–1753 (Sep. 2015)

C&Cトラフィック分類のための機械学習手法の評価

山内 一将[1,2,†1] 川本 淳平[1,2] 堀 良彰[2,3,a)] 櫻井 幸一[1,2]

受付日 2014年12月8日, 採録日 2015年6月5日

概要：インターネットの普及にともない，ボットネットによる被害が増大している．一般的なボットネットでは踏み台となる端末の制御を行うために Command and Control（C&C）サーバを利用している．そのため，ボットネット対策手法の 1 つとして C&C サーバの検知が注目されている．しかし，C&C サーバが用いるプロトコルの多様化により通信方法やネットワーク構造も多様化し，C&C サーバの検知が困難となっている．本研究では C&C サーバが用いるプロトコルによらず C&C サーバを検知するために特徴ベクトルの定義を行う．また，実データを用いて通常の通信と C&C サーバによる通信の分類を行い，C&C サーバの用いるプロトコルに特化しない手法としての有効性を示す．

キーワード：ボットネット，C&C サーバ，異常検知，機械学習

Evaluation of Machine Learning Techniques for C&C Traffic Classification

Kazumasa Yamauchi[1,2,†1] Junpei Kawamoto[1,2] Yoshiaki Hori[2,3,a)] Kouichi Sakurai[1,2]

Received: December 8, 2014, Accepted: June 5, 2015

Abstract: With the spread of Internet, the number of damage from botnet is increasing. General botnet use Command and Control (C&C) server and detecting C&C server is one of the technique of botnet measures. However, it is hard to detect C&C server because of diversification of C&C protocol and changing of botnet configuration. In our work, we define a feature vector to detect C&C server and report the experiment result that is classification normal traffic and C&C session by using real network traffic. Finally we show the effectiveness as the method of detecting C&C server which use several kinds of protocols.

Keywords: botnet, C&C server, anomaly detection, machine learning

1. はじめに

近年インターネットの普及にともない，ボットネットによる脅威が問題化している．ボットネット対策を行うために多くの研究がなされており，中でも Command & Control（C&C）サーバを特定する手法が必要とされている [1]．C&C サーバは，攻撃者がボットネットを運用するために利用するサーバである．C&C サーバは攻撃者からの指令をボットに感染した端末群へ転送する．C&C サーバの中には感染した端末群へいっせいに指令を送る，いわば増幅器のような役割を果たしているものも確認されている．C&C サーバから命令を受けた端末群は指令の内容に従って，DDoS 攻撃，スパムメールの送信，脆弱性スキャン攻撃などを行う．つまり，C&C サーバ通信（C&C トラフィック）はボットネットによる攻撃の予兆として考えられており，攻撃を未然に防ぐための 1 つの手法として C&C サーバの特定が必要とされている．

1.1 既存研究

C&C サーバが利用するプロトコルは IRC，HTTP，P2P

1 九州大学
Kyushu University, Fukuoka 819–0935, Japan
2 公益財団法人九州先端科学技術研究所
Institute of Institute of Systems, Information Technologies and Nanotechnologies (ISIT), Fukuoka 814–0001, Japan
3 佐賀大学
Saga University, Saga 840–8502, Japan
†1 現在，西日本電信電話株式会社
Presently with NIPPON TELEGRAPH AND TELEPHONE WEST CORPORATION
a) horiyo@cc.saga-u.ac.jp

などに分類できる．本稿では，簡略化のためにIRCを利用したボットネットをIRC型ボットネットと呼ぶ．同様にHTTP，P2Pに関してもHTTP型ボットネット，P2P型ボットネットと呼ぶ．

IRC型ボットネットは1993年頃から用いられてきたボットネットである [1]．IRC型ボットネットにおけるC&Cサーバの検知手法として，文献 [2], [3] がある．文献 [2] では，IRCクライアントがC&Cサーバと行う通信に関して，セッション情報から得られる特徴ベクトル，最初の16パケットの列を考慮した特徴ベクトル，パケットのデータサイズとパケット送信時間の間隔をヒストグラムで表現した特徴ベクトルの3種類を定義している．これらの特徴ベクトルに関して，機械学習を用いて分類した検知率，誤検知率，見逃し率の比較，評価を行っている．文献 [3] では，IRCプロトコルの通信特性に基づいてボット検知を行っている．著者らは，IRCプロトコルを用いた通信特性を見つけるために，スコア関数やブラックリスト/ホワイトリスト方式とn-gramによる分析を組み合わせている．

また，2003年頃からはボットネット制御の中心となるC&Cサーバを必要としないP2P型ボットネットが出現している [1]．P2P型ボットネットでは端末間で直接通信を行い，攻撃者からの指令は端末どうしで共有して拡散されるため，ボットネットの攻撃の対策が困難である．P2P型ボットネット対策に関する既存研究として，PeerShark [4] がある．文献 [4] では，ボットネットを他の通信と区別するために，通信の頻度やデータサイズなどを基に分類を行う手法を提案している．

IRC型ボットネットやP2P型ボットネットに比べて新しいボットネットとしてHTTP型ボットネットがある．HTTP型ボットネットは2005年頃に確認されている [1]．HTTPはIRCに比べると普及しているプロトコルであり，HTTPサーバに関するソフトウェアが充実していることからHTTP型ボットネットは近年増加傾向にある．HTTP型ボットネットの対策手法として，HTTPメソッドに着目した研究がある [5], [6]．HTTPプロトコルでは，データを取得する際のメソッドとしてGETやPOSTなどが利用される．そこで，これらHTTPメソッドを送信する時間に着目して，著者らが提案した方式によりクラスタリングを行っている [5]．文献 [6] では，Artificial Immune System（AIS）[10] を用いたリアルタイム検知を行う手法を提案している．一般的なAISは生物の免疫系の原理やプロセスをモデル化したものであり，この概念を著者らの手法に組み込むことでHTTP型ボットネットの検知をより効率的に行うことができると述べている．

1.2 研究課題と貢献

ボットネットの構造や利用するプロトコルが多様化しているため，これらの変化に対応した手法が必要である．文献 [3], [4], [5] はそれぞれ1つのプロトコルに特化した検知手法となっている．そのため，網羅的なボットネット検知が困難であり，また新しいプロトコルを利用したボットネットの検知も難しい．また，ボットネット対策手法の実用化に向けて実データを用いた評価が重要であるが，文献 [5], [6] ではそれらの評価ができていない．また，文献 [3] ではIRCプロトコルで用いられる，ニックネームと呼ばれる識別子を利用したシグネチャベースの検知を行っているため，未知のシグネチャに対する検知が難しい．

本研究ではプロトコルを仮定しないボットネット検知手法を提案する．我々が行った貢献は2つある．

- C&Cトラフィック特性に関する調査
  攻撃者は正規のIRCサーバやHTTPサーバなどをC&Cサーバとして悪用する．C&Cサーバはボットに感染した端末へ命令を送る．本研究では，パケットヘッダ情報を利用してトラフィックの特性調査を行った．
- C&Cトラフィック抽出実験
  実ネットワークを流れる通信データにおいて，C&Cトラフィックは通常の通信に紛れている．そのため，通常のトラフィックとC&Cトラフィックを分類する必要がある．そこで前項で得られたC&Cトラフィックの特性を元に特徴ベクトルを定義し，機械学習の分類アルゴリズムを用いてC&Cトラフィックの抽出を行う．

本稿の構成は，まず2章でボットネットについて述べ，3章でC&Cトラフィックの特性調査を行う．そして，4章で評価実験を行い，5章で実験結果と考察，6章で機械学習手法の比較について述べて，7章で結論とする．

2. ボットネット

ボットネットは悪意のある活動を目的としたネットワークのことであり，攻撃者が第三者のコンピュータに悪性プログラムを忍び込ませることにより作成された複数のボットと制御を司るC&Cサーバによって構成される．規模としては小さいものでは数十台，大きいものでは数十万台ものボットで構成される．本章ではボットネットが行う挙動に関して時系列調査を行う．

ボットネットの攻撃の挙動に関して**図 1** に示す．ここでは1台のハニーポットがマルウェアに感染してから攻撃に至るまでを時系列で示しており，CCCDATASet'10で取得されたハニーポット上で実行されたマルウェアのトラフィックに関して調査することで図1に示す挙動を観測した．図1ではマルウェアダウンロード，C&Cサーバとの通信，攻撃という3つのフェーズに分け，午前1時から午前4時までを観測している．まずマルウェアダウンロードを，ボットネットが一般ユーザに対して脆弱性探索に成功した場合に行う．ダウンロード時間に関しては，3回の

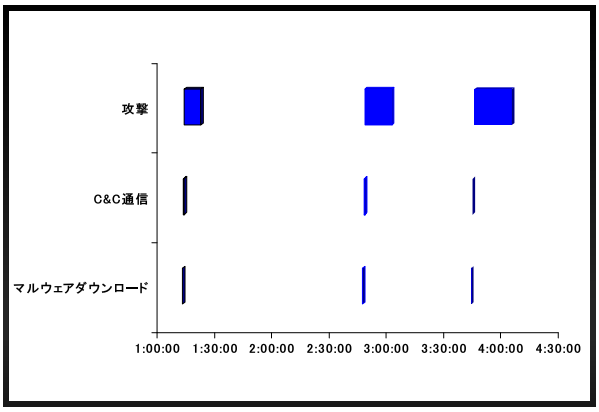

図 1 ボットネット活動の時系列調査
Fig. 1 Time-chart of Botnet activity.

平均で 3 秒程度であった．次にマルウェアに感染してボット化した PC は C&C サーバと通信を行う．通信時間は 3 回の平均で 19 秒程度であった．C&C サーバから指令をもらったボットは攻撃を行う．攻撃の時間としては 3 回の平均で 14 分 7 秒程度であった．このように，ボットネットが段階的に攻撃へ至るまでには何らかの予兆が観測される．その予兆の 1 つとして C&C サーバとの通信がある．C&C サーバとの通信を検出して通信を遮断することで，C&C サーバに接続されているボットからの攻撃を未然に防ぐことが可能である．

3. C&C トラフィック特性調査

本研究では C&C トラフィック特性調査を行うために CCCDataSet'09（以下，C09），CCCDataSet（以下，C10），PRACTICE'13（以下，P13）を用いた．これらのデータ・セットはサイバークリーンセンターに設置されているハニーポットで収集しているボットの観測データである [7]．本稿ではこれらのデータに含まれる IRC と HTTP を利用した C&C サーバ特性調査について述べ，その結果を基に特徴ベクトルを設計する．

3.1 ボットネットの通信形態

ボットネットを制御する C&C サーバはボットとの通信手段として大きく 2 つの形態を取る．1 つ目は，C&C サーバからボットへの一方向的な通信である．これは IRC を利用する場合に主に起こる．IRC をベースにしたボットネットは攻撃者に従来用いられてきた手法であり，これに関する既存研究も 1.1 節に示したように多くある．2 つ目は C&C サーバとボットの両方向の通信である．これは HTTP や P2P などを利用する場合に行われる．特に HTTP に関して，IRC よりも一般ユーザに普及しているプロトコルであるため HTTP 型ボットネットは増加傾向にある．HTTP の通信量は IRC に比べて大きく，異常な通信のみを正確に取り出すことが困難な傾向にある．

表 1 特徴ベクトル
Table 1 Feature vector.

V_1	送信パケット数（PKT）
V_2	送信データサイズ（Byte）
V_3	受信パケット数（PKT）
V_4	受信データサイズ（Byte）
V_5	セッション時間（s）
V_6	アクセス回数（回）
V_7	アクセス時間標準偏差

3.2 特徴ベクトルの定義

本節では特徴ベクトルを定義するために，C&C サーバの通信特性について調べる．文献 [8] ではマルウェア検知のための特徴量として 36 個の要素を選択し，正常トラフィックと感染トラフィックの分離を行っている．しかし，文献 [8] では通信方向の区別を行っていない．C&C サーバとボットの通信を取り出す場合には送信，受信の双方向通信に着目することが有効である．たとえば，IRC サーバにおいてあるクライアントがチャットをするためのグループ構成で必要となるチャンネルに参加した場合に，一種の通常な通信としてクライアント側ではパケットを送信，受信しながらメッセージを交換することが考えられる．一方で，ボットがチャンネルに参加する場合はユーザの意図していない部分での挙動となるため，チャンネルに参加したボットは自発的に C&C サーバに向けてメッセージを送ることが考えにくい．そのため，C&C サーバからボットに対してメッセージを一方的に受信することが考えられる．ゆえに本研究で定義する特徴ベクトルには双方向の通信を考慮したものを定義する．

また，文献 [5] では HTTP 型ボットが C&C サーバへアクセスする挙動の周期性に着目していた．しかし，周期的なアクセスは HTTP 型ボットネットに限ったことではなく DNS や P2P でも同じことがいえると考えられる．また，文献 [2] では IRC 型ボットネットを検知するためパケット数，パケットサイズなどを考慮した特徴ベクトルを定義していたが，アクセス挙動に関する属性が含まれていない．ここでのアクセスとはクライアントがサーバに TCP を用いて接続することを指し，アクセス挙動を考慮した特徴ベクトルを定義することが本研究において有効である．

したがって，我々は特徴ベクトルを**表 1** の要素を用いて定義する．本研究ではサーバ/クライアント通信に着目し，クライアントがサーバに TCP を用いて接続してから，接続を切るまでの通信を 1 つのセッションとして解析する．また，V_6, V_7 ではアクセス挙動特性を考慮した属性を定義している．特徴ベクトルの要素に関して説明する．V_1，V_2 はそれぞれボット（クライアント）が C&C（IRC または HTTP）サーバへ 1 回のセッションで送ったパケット，データサイズの総数を指す．同様に，V_3，V_4 ではそれぞれボットが C&C サーバから受信したパケット，データサイズの総数を指す．

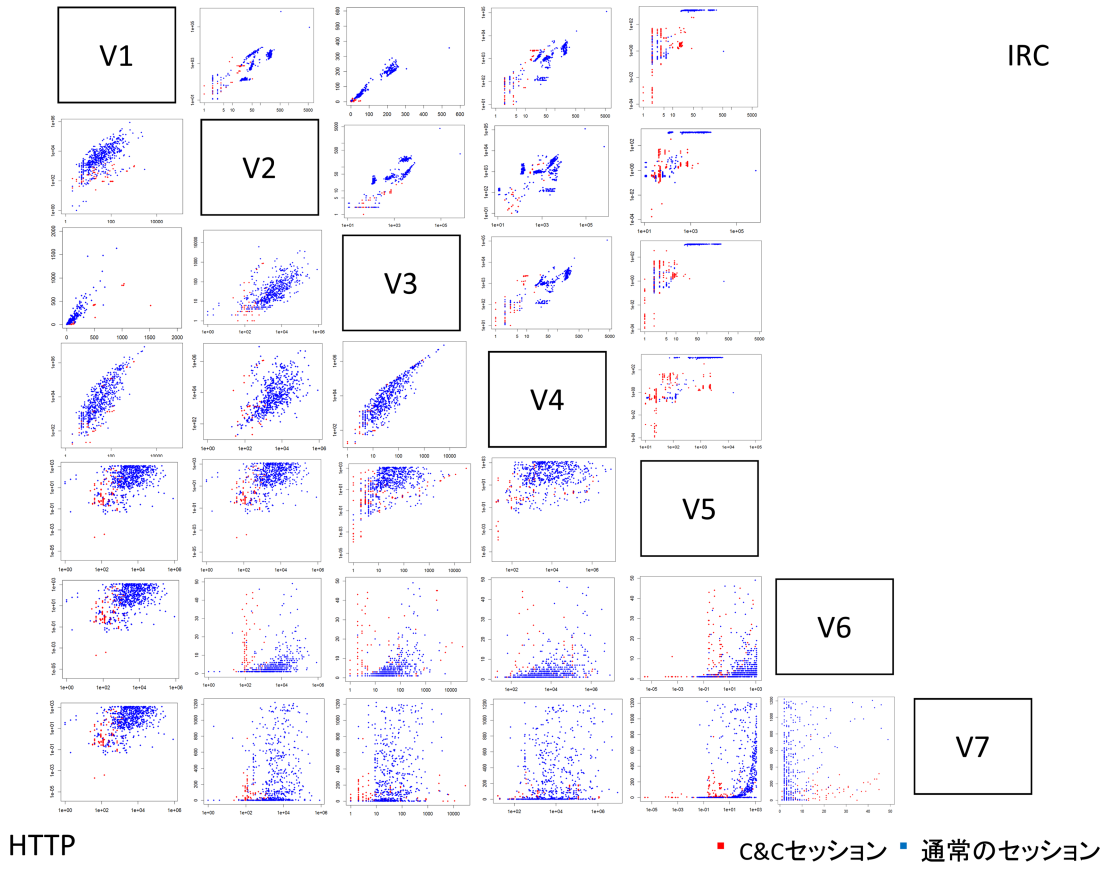

図 2 C&C セッション分析（全結果）
Fig. 2 C&C session analysis (All).

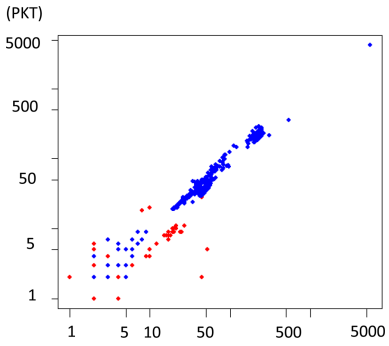

図 3 $V_1 - V_3(IRC)$
Fig. 3 $V_1 - V_3(IRC)$.

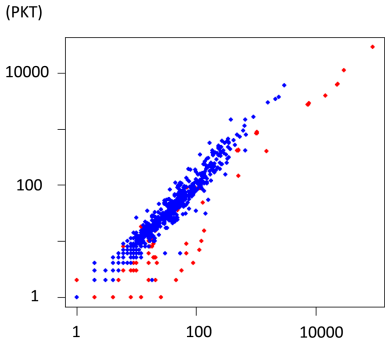

図 4 $V_1 - V_3(HTTP)$
Fig. 4 $V_1 - V_3(HTTP)$.

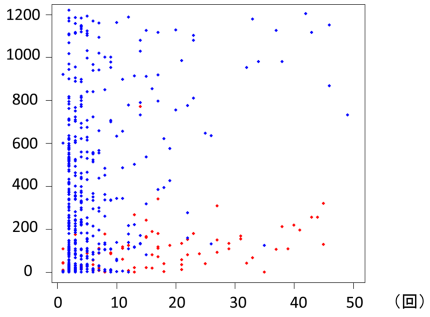

図 5 $V_6 - V_7(HTTP)$
Fig. 5 $V_6 - V_7(HTTP)$.

データサイズの総数に関しては，パケットのヘッダ情報を基にセッションごとに含まれているパケットのデータサイズを合計したものである．V_5 はパケットのヘッダ情報に含まれるタイムスタンプを確認し，セッション終了時刻からセッション開始時刻の差をとった時間である．また，V_6 はセッション中にクライアントがサーバへアクセスする回数の合計を指し，V_7 はアクセス時間のばらつきを表している．

3.3 C&C セッション分析

本節では通常のセッションと C&C セッションが提案する特徴ベクトルで分類可能であるか分析を行う．2 章より，C&C サーバの通信はボットネットが攻撃を行う予兆の 1 つとして考えることができ，C&C セッションを検出することでボットネットによる攻撃を未然に防ぐことを可能にする．**図 2** は IRC と HTTP の通信に関してそれぞれ特徴ベクトルを用いて解析を行った結果を示しており．通常の HTTP または IRC セッションは青で，C&C セッションは赤で示している．また，IRC に関しては，セッション中に C&C サーバへの再接続を行わないので V_6，V_7 に関しては考慮しない．図 2 から，IRC の方がデータの分布範囲が狭いことが分かる．これに対し，HTTP ではデータの分布範囲が広く，IRC よりも通信の多様性が見られる．

図 2 の結果において，特に 2 種類のデータを区別できた結果に関して抜粋したものを**図 3**，**図 4**，**図 5** に示す．図 3

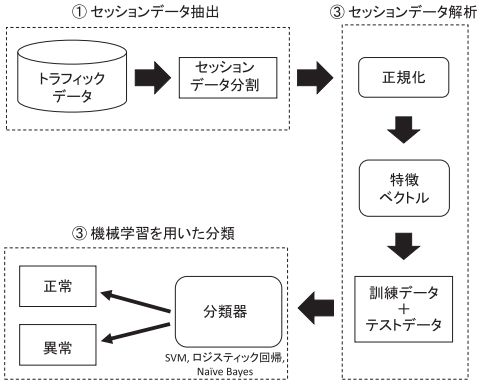

図 6 実験の流れ
Fig. 6 Experiment flow.

に関しては，ほとんどの IRC セッションが送信パケット数，受信パケット数ともに 25 から 500 に集中しており，割合はおよそ 1 対 1 であるデータが多い．これに対し C&C セッションでは送信パケット数，受信パケット数が比較的少なく，割合はおよそ 1 対 1 から外れるものが多い．図 4 に関しては，ほとんどの HTTP セッションと C&C セッションは送信パケット数，受信パケット数の双方で 5 から 10,000 までに幅広く分布している．しかし，HTTP セッションはおよそ 1 対 1 の割合で分布しているのに対して C&C セッションはおよそ 1 対 1 から外れるものが多い．図 5 に関しては，HTTP セッションではアクセス回数に関係なくアクセス時間間隔の標準偏差が大きいことが分かる．これに対して，C&C セッションではアクセス回数が増えてもアクセス時間の標準偏差が急激に増えることがない．

これらの結果より，我々が提案する特徴ベクトルを用いたセッション分類を高い検知率，低い誤検知率で実現できると考えられる．次の章では実データを用いた評価実験を行う．

4. 評価実験

C&C トラフィックを検出するために，実データを用いた実験についての説明を行う．**図 6** は，今回行った実験の流れを示している．本章では我々が行った実験に関してセッションデータ抽出，セッションデータ解析，機械学習を用いた分類に関してそれぞれ説明する．

4.1 セッションデータ抽出

今回の実験ではあらかじめ採取されているデータを利用する．そのため，正常，異常と判断することのできるデータを初めに用意する．データ収集の方法として，Linux コマンドライン上でパケットを観測可能なツールである tcpdump を利用している．正常なデータと異常なデータでは採取時期や取得方法が異なるので，本節ではそれぞれのデータ取得方法に関して説明する．

表 2 ユニーク IP アドレス数
Table 2 Number of unique IP address.

	Normal	C&C		
		C09	C10	P13
IRC	736	6	19	0
HTTP	763	51	139	15
計	1,499	57	158	15

表 3 抽出したセッションデータの数
Table 3 Number of extracted session data.

	Normal	C&C		
		C09	C10	P13
IRC	903	190	573	0
HTTP	1,270	84	255	406
計	2,173	274	828	406

4.1.1 正常なデータ

正常なデータは我々の研究室にあるサーバを監視することで収集する．採取期間は 2012 年 8 月から 9 月で今回は大学内のネットワークは安全であると仮定している．そして，採取されるデータのポート番号に着目し，6667 番を利用しているものは IRC，80 番を利用しているものは HTTP の通信とする．

4.1.2 異常なデータ

異常なデータは，C&C サーバとボットの通信を含んだデータを指す．異常なデータを取得するために，本研究では C09，C10，P13 を用いる．これらのデータ・セットはハニーポット上で実行されたマルウェアのデータを収集したものであり，これらのデータから目的としている通信を取り出すことを考える．そのために，我々はパケットペイロード部の情報を精査する．具体的には，クライアントが C&C サーバへ接続する際に用いるコマンドに着目する．一般的に IRC では JOIN コマンドを用いてサーバへログインし，HTTP では GET コマンドを用いてサーバにデータを要求する．我々は，ボットと C&C サーバで同じ状況を想定し，JOIN，GET を使っている通信に関して取り出す．

このようにして採取されたデータに関して**表 2**，**表 3** ではそれぞれユニーク IP アドレス数，抽出したセッションデータの数について示している．P13 に関しては上記の条件で抽出を行ったが，IRC トラフィックは検出されなかった．

4.2 セッションデータ解析

次にセッションデータの解析を行い，それらを数値列ベクトルとして出力させる．本研究では表 1 に示した特徴ベクトルを利用する．本手法で設計した特徴ベクトルでは，予兆検知を行うために C&C トラフィックを識別することを目的としており，TCP コネクションが確立される通信に対して構成可能であり，初めて TCP コネクションを確

立してから 20 分間に新たな TCP コネクションの確立があるかについての精査を行う．そのため，TCP コネクションを確立していない通信に関しては特徴ベクトルが構成されない．

表 4，**表 5** には IRC，HTTP それぞれの解析結果で得られた平均値と分散値について示す．表 4 に関して，C&C に比べて IRC の方が V_1 から V_5 の平均値，分散値が小さいことが分かる．また，C10 の V_1 から V_5 に関して C09 と近い平均値を得ることができたが分散値は大きいことがわかる．IRC セッションの場合は，一回のセッションでサーバへの再接続を行わないため $V_6 = 1$，$V_7 = 0$ と一意に決まる．表 5 に関して，C&C の方が HTTP よりも V_6 以外の平均値が小さい．また，P13 に関して，V_1 から V_7 の分散値が小さい．C09，C10 に関しては V_4 の分散値が大きい．このことから，受信データサイズは C&C の場合では大きいものがあることが分かる．HTTP は IRC に比べてセッションデータの形態が多様であることも分かる．

次に，定義した特徴ベクトルを用いてセッション解析した結果に関して解析データの管理を簡素化するためにデータの正規化を行う．i 番目のセッションデータに関して，j 番目の属性値を正規化したい場合，次の式で表すことができる．

$$\hat{x}_{i,j} = (x_{i,j} - \min(x_{n,j}))/\max(x_{m,j})$$

ここで，j 番目の属性値に関して n 番目のセッションデータ $x_{n,j}$ で最小値を，m 番目のセッションデータ $x_{m,j}$ で最大値を取る．また，i 番目のセッションデータで j 番目の属性値 $x_{i,j}$ を正規化した値は $\hat{x}_{i,j}$ とする．これにより，各データの特徴量は最小値が 0，最大値 1 の実数値となる．

表 4 IRC セッションデータ解析結果：平均値（分散値）
Table 4 IRC session data analysis: Average (variance).

	Normal (IRC)	C&C	
		C09	C10
V_1	88 (6.0×10^3)	6 (24)	5 (250)
V_2	1,187 (3.6×10^6)	67 (1.5×10^4)	77 (1.9×10^4)
V_3	75 (6.1×10^3)	2 (6.9)	3 (632)
V_4	1,336 (2.2×10^6)	177 (1.7×10^5)	185 (1.6×10^6)
V_5	583 (2.8×10^5)	8 (75)	6 (111)
V_6	1 (0)	1 (0)	1 (0)
V_7	0 (0)	0 (0)	0 (0)

4.3 機械学習を用いた分類

本節では機械学習による分類について説明する．今回の実験では教師あり学習として用いられている 3 つの識別モデル，SVM，ナイーブベイズ，ロジスティック回帰を適応する．文献 [2] では IRC 型ボットネットでの C&C トラフィックを抽出するために複数の識別モデルでの比較を行うことで SVM の有効性を示している．しかし，今回の実験では IRC，HTTP の両方を扱うことに加えて，3.3 節での C&C セッション分析から HTTP 通信の挙動パターンは IRC に比べて多いことが分かったので，再度複数の識別モデルを用いた評価が必要となる．これらのモデルは教師あり学習であり，学習データと呼ばれる入出力のペアの事例が複数与えられているデータが必要となる．それを基に，テストデータの新しい入力データに関しても正しい出力ができることを目的としたものが教師あり学習ある．我々は，ボットネット対策のための実用的なシステムとしてどの識別モデルを使うことが適切であるかに関して分類精度と実行時間から評価を行う．機械学習の機能を実現させるために我々は R [13] を利用している．

機械学習のモジュールに関して，SVM では kernlab パッケージ [14]，ロジスティック回帰では glmnet パッケージ [15]，ナイーブベイズ法では e1071 パッケージ [16] を利用している．また，SVM 適応の際のカーネル関数はラジアル基底関数

$$k(\vec{x}, \vec{y}) = \exp \frac{-||\vec{x} - \vec{y}||^2}{2\sigma^2}$$

を用い，予備実験により適切と思われる σ を設定している．また，SVM とロジスティック回帰では学習データから最適なチューニングパラメータを決定するために交差検定を行う．今回の実験では 3 回の交差検定を行っている．ナイーブベイズに関しては，今回の実験で最適化するパラメータがなかったため，交差検定を行っていない．

今回の実験で用いた学習データとテストデータに関して説明する．正常なデータ，異常なデータからデータ・セッ

表 5 HTTP セッションデータ解析結果：平均値（分散値）
Table 5 HTTP session data analysis: Average (variance).

	Normal (HTTP)	C&C		
		C09	C10	P13
V_1	88 (1.5×10^7)	60 (1.4×10^2)	47 (1.3×10^4)	4 (5.7)
V_2	33,140 (3.9×10^9)	194 (5.7×10^2)	177 (2.1×10^9)	126 (1.4×10^2)
V_3	129 (1.7×10^6)	50 (900)	35.4 (1.4×10^7)	3.4 (74)
V_4	33,671 (2.1×10^{12})	66,320 (1.9×10^9)	42,212 (2.6×10^9)	1,135 (1.1×10^4)
V_5	249 (1.2×10^5)	2.6 (2.8)	0.27 (1.3×10^4)	1.7 (3.7)
V_6	9.15 (1.3×10^6)	3.8 (0.13)	35.7 (7.8×10^5)	1.1 (0.3)
V_7	122 (1.3×10^5)	0.64 (2.3)	3.1 (6.67)	1.5 (0.5)

トの種類に関係なくランダムに 2/3 のセッションデータを取り出して学習データに，残りの 1/3 をテストデータに割り当てている．これにより，すべてのパターンの通信に対してテストデータの分類が可能であるかの評価を行った．

5. 実験結果と考察

今回の実験では，すべての通信パターンを学習したときに，提案した特徴ベクトル（以下，提案ベクトル）によって正確にセッション分類が可能であるかの評価実験を行った．実験結果を**図 7** に示す．図 7 では HTTP，IRC のセッションに関しての分類結果を検知率，誤検知率で表している．

HTTP に関して，提案ベクトルでは既存の特徴ベクトル [2]（以下，既存ベクトル）と比べて検知率が SVM を用いた場合 22.3%，ロジスティック回帰（以下，図 7 では LR と記載）の場合 8.2%，ナイーブベイズ（以下，図 7 では NB と記載）の場合では 3.9%高くなり，誤検知率は SVM の場合 2.7%，ロジスティック回帰の場合 14.9%，ナイーブベイズの場合では 23.2%低くなった．このことから，アクセス挙動特性を考慮した特徴ベクトルである V_6，V_7 が分類精度向上に大きく貢献していることが分かる．ボットが C&C サーバにアクセスする際には，ボット自体がダウンロードしたマルウェアに従って機械的に動くため，正規ユーザによるアクセスとは異なる特徴が現れる．つまり，ボットの定期的なアクセスに対して正規ユーザはランダムな時間間隔でのアクセスを行うために通常のトラフィックと C&C トラフィックは分類できたと考えられる．しかし，正規ユーザの通信の中にはユーザのクリック挙動とは独立に，Web ブラウザ内に実装されているアプリケーションの通信が含まれている場合がある．たとえば，Ajax はそのようなアプリケーション実装技術の 1 つであり，非同期通信を利用したデータ取得や，動的な Web ページ更新などを可能にする．そのような通信に関してはアクセスの周期性をともなう場合があり，本手法を適応した場合に誤検知を起こす可能性が高いと考えられる．

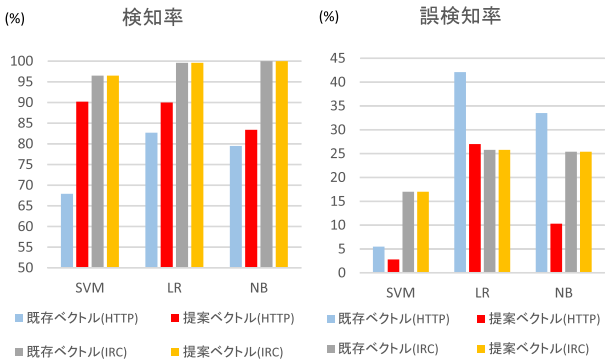

図 7 提案ベクトル，既存ベクトルとの比較に関する実験結果
Fig. 7 Comparison between proposed vector and existing vector.

IRC では，既存ベクトルと同精度の検知率，誤検知率を得た．IRC 通信において一度のセッションを行う際に再度アクセスを行わないので，$V_6 = 1$，$V_7 = 0$ で値が一意に決まり，既存ベクトルと同様 V_1 から V_5 までの特徴ベクトルを考えるからである．しかし，誤検知率の高さが課題点であり SVM を適応した場合の 17%が最小であった．誤検知率が高い問題を解決する方法としては，特徴ベクトルに新たな属性を付け加えることや，ホワイトリスト/ブラックリスト方式と本手法を連携させることなどが考えられる．

6. 機械学習手法の比較

5 章で述べたように既存ベクトルと比べて提案ベクトルを用いた分類を行う場合には，HTTP 通信では分類精度が向上し，IRC 通信に対しては同精度の分類精度が得られることが分かった．本章では評価実験で用いた SVM，ロジスティック回帰（以下，**表 6**，**表 7** では LR と記載），ナイーブベイズ（以下，表 6，表 7 では NB と記載）の機械学習手法について，性能評価を分類精度，実行時間の 2 点から行う．

6.1 分類精度

表 6 は提案する特徴ベクトルを用いた場合のデータ・セットごとの分類結果を示す．すべての機械学習手法に共通することとして，P13 に関する C&C セッションの検知がほぼできているという点である．これは P13 での HTTP サーバの通信パターンが少なかったために，P13 の一部の

表 6 データ・セットごとに見た実験結果
Table 6 Result of classifying every DataSet.

	Normal		C&C				
	IRC	HTTP	C09		C10		P13
			IRC	HTTP	IRC	HTTP	HTTP
SVM (Normal)	219	414	1	0	9	1	1
SVM (Anomaly)	45	12	66	32	208	100	103
LR (Normal)	196	251	0	2	1	14	0
LR (Anomaly)	68	115	67	30	216	87	104
NB (Normal)	197	382	0	8	0	17	0
NB (Anomaly)	67	44	67	27	217	84	104
計	264	426	67	32	217	101	104
検知率 (SVM) [%]			98.5	100	95,9	99.0	99.0
誤検知率 (SVM) [%]	17.0	2.8					
見逃し率 (SVM) [%]			1.5	0	4.1	1.0	1.0
検知率 (LR) [%]			100	93.7	99.5	86.1	100
誤検知率 (LR) [%]	25.8	27.0					
見逃し率 (LR) [%]			0	6.3	0.5	13.9	0
検知率 (NB) [%]			100	84.4	100	83.2	100
誤検知率 (NB) [%]	25.4	10.3					
見逃し率 (NB) [%]			0	15.6	0	16.8	0

表 7 機械学習アルゴリズムの実行時間の比較
Table 7 Comparison of execution time of machine learning algorithms.

	SVM	LR	NB
学習データ (s)	1.97	2.01	0.03
テストデータ (s)	0.12	0.81	0.38
合計 (s)	2.09	2.82	0.41

IRC では，既存ベクトルと同精度の検知率，誤検知率を得た．IRC 通信において一度のセッションを行う際に再度アクセスを行わないので，$V_6 = 1$，$V_7 = 0$ で値が一意に決まり，既存ベクトルと同様 V_1 から V_5 までの特徴ベクトルを考えるからである．しかし，誤検知率の高さが課題点であり SVM を適応した場合の 17%が最小であった．誤検知率が高い問題を解決する方法としては，特徴ベクトルに新たな属性を付け加えることや，ホワイトリスト/ブラックリスト方式と本手法を連携させることなどが考えられる．

データを学習することで分類できたと考えられる．識別モデルごとに見ると，SVM を利用した場合，C&C セッションの見逃しは全体的に少ないが，IRC では誤検知率 17.0%となった．ロジスティック回帰を利用した場合，C10 の HTTP に関して，見逃し率 13.9%となった．また，IRC，HTTP では 25.8%，27.0%の誤検知があることが分かる．ナイーブベイズ法を利用した場合，C09 の HTTP では見逃し率 25%，C10 の HTTP では見逃し率 16.8%であった．また，IRC では誤検知率 25.4%であることが分かる．

6.2 実行時間

表 7 では実験 A において提案方式を用いた場合の学習データでのモデル予測時間とテストデータの分類時間を示している．結果としてナイーブベイズでの学習データの読み込み時間が最も早く，SVM でのテストデータの分類時間が最も早いことが分かった．SVM，ロジスティック回帰では交差検定法により学習データから最適なパラメータチューニングを行ったのでナイーブベイズに比べて時間を要している．

6.3 性能評価

6.1 節，6.2 節の結果からから，ロジスティック回帰，ナイーブベイズでは網羅的に C&C セッションを検知し，誤検知が高くなる．特にナイーブベイズの場合には学習データの予測時間，テストデータの分類時間が高速であり，リアルタイム検知に適していると考えられる．実用面を考慮する場合，C&C トラフィック自体は攻撃を受ける前の予兆であるため，攻撃を受ける可能性がある場合にアラートをあげるようなシステムを想定するときは攻撃を受けるという信頼性は低くなるが，安全性を重視したい場合には有効であると考えられる．しかし，誤検知率が高くなるに連れて誤ったアラートを発する頻度が増えるので攻撃対策の信頼性やコストなどを考慮すると実用的ではない．また，C&C トラフィックをいかに早く検知し，対策を行うかという点も重要であるため，機械学習アルゴリズムの実行時間を考慮することも重要である．ゆえに，SVM を用いた場合に検知率が 90%以上で誤検知率も 4%程度という結果となっており，最も良い分類精度を得ていることに加えて，テストデータの分類時間もロジスティック回帰とナイーブベイズと比較して高速であるため，実用的であると考えられる．

7. 結論

本研究では，多様なプロトコルを利用する C&C サーバを特定するための新たな特徴ベクトルを提案した．この特徴ベクトルはクライアントとサーバの通信を区別することに加えて，クライアントがサーバへアクセスする挙動を表現したものとして定義した．この手法を用いることで C&C サーバが用いるプロトコルに特化せずに C&C トラフィックが検知可能な手法となることを示した．また，機械学習手法の分類精度に関して検知率はロジスティック回帰とナイーブベイズが高いが，誤検知率は SVM が低いことが分かった．また，実行時間に関して，学習データの読み込みはナイーブベイズが最も高速でテストデータの分類は SVM が最も高速であることが分かった．今後の課題として，今回扱った実データで用いられていない DNS，P2P などのトラフィックでの評価実験を行う．また，実用化に向けて，本実験での一連の流れを自動化するようなシステムの設計を行いたいと考えている．

謝辞 この研究の一部は，「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] Vania, J., Meniya, A. and Jethva, H.B.: A Review on Botnet and Detection Technique, *International Journal of Computer Trends and Technology*, Vol.4, No.1, pp.23–29 (2013).
[2] Kondo, S. and Sato, N.: Botnet Traffic Detection Techniques by C&C Session Classification Using SVM, *Proc. 2nd International Workshop on Security* (*IWSEC 2007*), pp.91–104 (2007).
[3] Goebel, J. and Holz, T.: Rishi: Identify bot contaminated hosts by IRC nickname evaluation, *Proc. 1st USENIX HotBots* (2007).
[4] Narang, P., Ray, S., Hota, C. and Venkatakrishnan, V.: PeerShark-Detecting Peer-to-Peer Botnets by Tracking Conversations, *Proc. IEEE Security & Privacy Workshops* (*SPW 2014*), pp.108–115 (2014).
[5] Ashley, D.: An Algorithm for HTTP Bot Detection, Research paper, University of Texas - Information Security Office (2011).
[6] Tyagi, A.K. and Nayeem, S.: Detecting HTTP Botnet using Artificial Immune System, *International Journal of Applied Information Systems*, Vol.2, No.6, pp.34–37 (2012).
[7] マルウェア対策研究人材育成ワークショップ 2014 (MWS2014)，入手先 ⟨http://www.iwsec.org/mws/2014/about.html⟩（参照 2014-12-05）.
[8] 市野将嗣，市田達也，畑田充弘，小松尚久：トラフィックの時系列データを考慮した AdaBoost に基づくマルウェア感染検知手法，情報処理学会論文誌，Vol.53, No.9, pp.2062–2074 (2012).
[9] Gu, G., Perdisci, R., 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 2008*) (2008).
[10] Castro, L.N. and Timmis, J.: *Artificial Immune Systems, A New Computational Intelligence Approach*, Springer (2002).
[11] Schehlmann, L. and Baier, H.: COFFEE: A Concept based on OpenFlow to Filter and Erase Events of Botnet activity at high-speed nodes, *Proc. INFORMATIK 2013*, pp.2225–2239 (2013).
[12] Gu, G., Perdisci, R., Zhang, J. and Lee, W.: BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection, *Proc. 17th USENIX Security Symposium* (2008).

[13] R project, available from ⟨http://www.r-project.org/⟩ (accessed 2014-11-10).
[14] Package 'kernlab', available from ⟨http://cran.r-project.org/web/packages/kernlab/kernlab.pdf⟩ (accessed 2014-11-10).
[15] Package 'glmnet', available from ⟨http://cran.r-project.org/web/packages/glmnet/glmnet.pdf⟩ (accessed 2014-11-10).
[16] Package 'e1071', available from ⟨http://cran.r-project.org/web/packages/e1071/e1071.pdf⟩ (accessed 2014-11-10).

山内 一将

2013 年九州大学工学部電気情報工学科卒業．2015 年九州大学大学院システム情報科学府情報学専攻修士課程修了．修士（工学）．同年西日本電信電話株式会社入社．

川本 淳平 （正会員）

2007 年京都大学工学部情報学科卒業．2012 年同大学大学院情報学研究科博士後期課程修了．京都大学博士（情報学）．情報通信研究機構有期研究員，筑波大学システム情報系研究員を経て 2013 年より九州大学大学院システム情報科学研究院助教，ならびに九州先端科学技術研究所特別研究員（兼務）．クラウドデータベースにおけるプライバシおよびプライバシ保護データマイニングの研究に従事．IEEE，ACM，電子情報通信学会，日本データベース学会，人工知能学会各会員．

堀 良彰 （正会員）

1992 年九州工業大学情報工学部電子情報工学科卒業．1994 年同大学院情報工学研究科情報システム専攻修士課程修了．1994 年九州芸術工科大学助手．博士（情報工学）．2004 年九州大学大学院システム情報科学研究院助教授．2013 年より佐賀大学全学教育機構教授．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の研究に従事．2000 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（現，公益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）特別研究員（兼務）．電子情報通信学会，ACM，IEEE，各会員．

櫻井 幸一 （正会員）

1988 年九州大学工学研究科応用物理学専攻修士課程修了．同年三菱電機（株）入社．現在，九州大学大学院システム情報科学研究院情報学部門教授．2004 年より九州システム情報技術研究所第 2 研究室（現，九州先端科学技術研究所・情報セキュリティ研究室）室長兼任．博士（工学）．2000 年情報処理学会坂井特別記念賞．2000 年，2004 年情報処理学会論文賞．2005 年 IPA 賞受賞．日本数学会，応用数理学会，電子情報通信学会，ACM，IEEE 各会員．