サポートベクターマシンを用いたC&Cセッションの特徴に基づいたHTTP型ボット検知

山内 一将[1]　堀 良彰[2]　櫻井 幸一[2]

概要：コンピュータウイルスによるユーザの被害件数が増大しており，問題となっている．その中でもボットネットと呼ばれる，遠隔操作によって攻撃を行う被害が深刻化している．本研究ではボットネットの活動を抑えるために，トラフィック観測により，C&C セッションの検出を行いたい．既存研究では IRC を利用したボットネットの検知手法はたくさんあるが，それに比べて HTTP を利用したボットネットの検知手法についての研究は少ない．そこで，本研究では HTTP を利用したボットネットに焦点を置く．また，通常の HTTP セッションと C&C セッションの分類を行うために，サポートベクターマシンを用いる

キーワード：コンピュータ・ウィルス，侵入検出・検知，分類学習

HTTP-based Botnet Detection Based on The Feature of C&C Session Using by Support Vector Machine

Abstract: Increasing of the user's damage number by computer virus, it has been problem. Even inside of it, it is more aggravaion the damage which is attacked by remote control, and the attacks is deliverd by the botnet. In my research, in order to suppress the botnet activity, we want to detect the C&C session by traffic observation. In the existing research, although there are many detection techniques focus on IRC-based botnet, there are little detection techniques focus on HTTP-based botnet compared with it. Therefore, we focus on HTTP-based botnet, and in order to classify normal HTTP session and C&C session, we use Support Vector Machine.

Keywords: computer virus, Invasion detection, classification study

1. 序論

インターネットは我々の生活の中で必要不可欠なものになっている．また，コンピュータの性能は年々向上しており，コンピュータの普及率も急激に増大している．技術的な開発が進む一方で，ユーザのセキュリティに対する意識が低いという問題点が挙げられる．特に，マルウェア感染の被害件数は増加が目立つ．マルウェアとは，悪意のあるプログラムの総称で，感染したコンピュータは，ユーザの個人情報を盗み出されたり，フィッシングページに移動させられるなどして，プライバシーの侵害や，金銭的な被害を受けることがある．以前のマルウェアを作る攻撃者は，金銭的な利益を得ることを目的としない，愉快犯が多かったが，近年は感染したコンピュータから得た個人情報やその他の情報を売ることで，金銭的な利益を得ることを目的としている場合が多くなった．情報化社会である現在において，漏洩した情報によっては，その国全体が危機に脅かされるようなものもあり，マルウェアは現代のインターネットセキュリティーを脅かすものの一つとなっている．

マルウェアの検知手法で用いられている手法として，パターンマッチング手法がある．この手法は，ホストベース，ネットワークベースの両方からの検知に従来用いられてきた信頼性の高い手法である．しかし，この手法は既存のマルウェアのバイナリーコードに依存しており，新規マルウェアには対応できていない．そのため，新規マルウェアが増加傾向にある現代において，パターンマッチング手

[1] 九州大学工学部電気情報工学科
Department of Electrical Engineering and Computer Science, Kyushu University
[2] 九州大学システム情報科学府情報学専攻
Department of Imformatics, Kyushu University

法には限界がある．過去のマルウェアの挙動から未知のものを予測で検知するヒューリスティック手法というものも存在するが，誤検知率が高いという欠点がある．そこで，新規マルウェアも高精度で検出可能な手法を用いてのボット検知が本研究の目標である．マルウェアの中でも特に，ボットによる被害が増加し，問題となっている．IP アドレス詐称による冤罪逮捕のような事件なども存在する．ボットは複数の端末でボットネットを形成する．ボットネットは数十台〜数万台のボットで構成されており，C&C サーバ (C&C) サーバという中継サーバを介して操作される．操作された複数台のボットはフィッシングメールの一斉送信や，特定のサーバへの一斉アクセス，などといった攻撃活動を行う．このような攻撃による被害を減らすためにも，ボットネットの活動を抑制する必要がある．

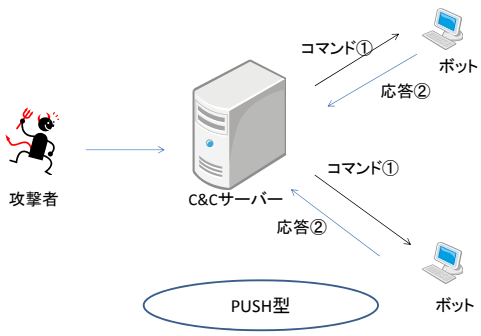
メールやソフトウェアからの感染のみならず，我々が日頃利用している Web サイトからのボットへの感染が目立っている．例えば，悪性なサイトへのアクセスや，メールやウェブ等に添付された不正ファイルを開くことなどによって感染してしまうことがある．コンピュータのアップデート，セキュリティーソフトの導入，ブロードバンドルーターの利用などを行なうような対策が見られるが，新規マルウェアの急激な増加により，それらの対策だけではコンピュータのボットへの感染から，完全に防ぐことができたとは言い難い．

本研究の目的としては，ボットネットの活動を抑えるための足がかりとして，ボットと C&C サーバの通信を検出することを目的とする．

2. ボットネット

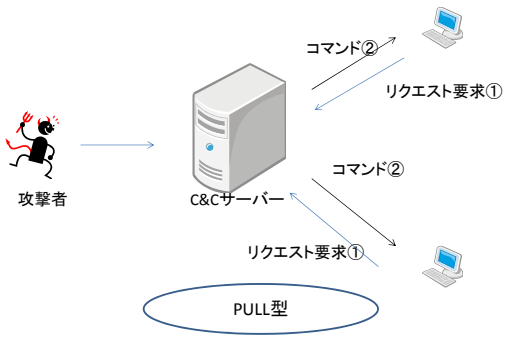
ボットネットとはボットによって構成されたネットワークのことであり，ハーダーが C&C サーバを介してボットに命令することで，ボットは攻撃を行う．ハーダーはボットネットを統括する攻撃者のことである．ハーダーが，C&C サーバを介してボットに命令を送ることで，Dos 攻撃や，スパムメールの送信，ポートスキャンなどの攻撃をボットに行わせる．

ボットネットの利用する通信形態はいくらかある．例えば，IRC，HTTP，P2P などである．IRC はインターネット上で利用されるチャットで Internet Relay Chat のことである．HTTP は，Web サーバとクライアントのデータのやり取りで使われる方式である．P2P とは Peer-to-Peer のことで，サーバを必要としない，端末同士でやり取りを行う方式である．C&C サーバとは，前述したように，ボットネットの通信を制御するためのサーバである．C&C サーバとボットの通信について見ると，IRC では PUSH 型の形態が取られており，HTTP では PULL 型の形態が取られている．PUSH 型では，ボットの意思に関係なく命令を受け取り，メールマガジンや電子メールはそれに相当する．PULL 型では，ボットが C&C サーバにリクエストを要求することで命令を受け取り，Yahoo や，Google 等の検索エンジンで検索を行う場合は，PULL 型に当たる．本研究は PULL 型の特性を利用した，HTTP 型ボットネットの検出を試みる．



21

図 1 IRC 型ボットネット



22

図 2 HTTP 型ボットネット

3. 既存研究

ボット検知手法については既存研究がいくらかある．この章では，その既存研究について挙げる．

3.1 シグネチャによるボット検知手法

1 章でも前述したように，ボットは日々亜種が発生しており，その度に解析をする必要があるため，未知のボットに対応することが難しい．

3.2 ブラックリスト方式によるボットの検知手法

有害であるとされた Web サイトなどのブラックリストを作成して，フィルタリングを実施する方式のことである．ブラックリストにリストアップされたアドレスは通信事業者がブロックする．このブラックリスト方式に対して，ホワイトリスト方式がある．ホワイトリスト方式では，安全なアドレスを集めたデータベースを作成し，そのデータベースに含まれていないものをブロックするという方式である．しかし，この方式も亜種が発生する度に，解析をする必要があるため，シグネチャによるボット検知手法と同様に，未知のボットに対応することが難しい．

3.3 ネットワークトラフィック観測による異常検知

ネットワーク上ではプライバシーの問題が重要視されているため，トラフィックデータのペイロードを見ずに，ヘッダ情報から様々な特徴量を抽出する．パケット数の累積，送信元 IP と宛先 IP の関係などをグラフ化して，ポートスキャンの方法などを研究した [6].

3.4 ボットと C&C サーバの通信に着目したボット検知手法

C&C サーバの振る舞いに着目した既存研究について 3.4.1，3.4.2 節で挙げる．

3.4.1 特徴ベクトルを用いたボットネットトラフィック検知手法

ボットネットトラフィックを検知するために，特徴ベクトルを用いた検知手法を用いている [1]．ここでは 3 つの独立した特徴ベクトルを定義しており，これらによって抽出された特徴ベクトルのベクトル列をデータとして抽出する．抽出したデータを機械学習を用いて，通常の HTTP セッションと C&C セッションの分類を行い分類精度について評価した．この研究では IRC 型ボットに焦点を置いたものとなっている．

3.4.2 アルゴリズムを利用した HTTP 型ボット検知手法

[2] HTTP 型ボットは攻撃を行うために C&C に命令を得る動作を行う．その得る動作には周期性があり，HTTP 型ボットと C&C サーバの通信時間間隔における標準偏差が，通常の通信に比べて小さくなることを利用した手法である．ただし，ボットがネットワークを遮断や，何らかの理由で C&C サーバから命令を受け取ることができなかった場合，通信時間間隔の周期性に反する時間が現れることがある．これらの通信時間間隔を k-means 法に代わる手法として，著者らが提案したアルゴリズムなどを用いてクラスタリングを行い，それによりボットの検知を行うという手法である．

4. 提案手法

提案手法としてサポートベクターマシンを用いた HTTP 型ボットの検知を提案する．そのために，トラフィック観測によりセッション毎にパケットを取り出す． 取り出したパケットを，通常の HTTP セッションと C&C セッションに分類することで，C&C セッションを検出したい．本研究では，C&C セッションを検出するために，C&C サーバの振る舞いに着目する．C&C サーバに着目する理由としては，感染したボットのみを検出するだけでなく複数のボットを制御している C&C サーバ自体を遮断すれば，ハーダーはボットネットを操作できなくなり，ボットネットからの攻撃を防ぐことができる，という理由からである．

5. 実験概要

5.1 実験環境

HTTP 型ボット検知を行うために，通常の HTTP セッションと，C&C セッションの収集を行った．トラフィック観測については，5.2.2 節で説明する．本研究において，これらのデータ収集で用いたツールとしては，tcpdump と Wireshark(Ver.1.6.2) である．

表 1 データ数
*() の中の数字はテストデータ数

	HTTP セッション数	C&C セッション数	合計
A	130 (30)	70 (20)	200 (50)
B	125(27)	83 (31)	208 (58)
C	200 (30)	80(30)	280 (60)
D	150 (50)	91 (41)	241 (91)

また，表 1 では，通常の HTTP セッション，C&C セッションのデータ数について示す．それぞれの特徴ベクトルの説明については次のセクションで行う．今回は，それぞれの特徴ベクトルで，独立して実験を行った．そのためデータ数が異なっている． またサポートベクターマシンについて，セッション分類で用いるサポートベクターマシンに関して，今回は SVM_light をツールとして用いた．

5.2 トラフィック観測について

本研究ではトラフィック観測を行い，セッション毎にトラフィックを分割した．セッションを分割する際に，20 分以上のセッションが確立されていた場合はタイムアウトとした．図 3 では，クライアントと Web サーバの TCP 通信を示した．TCP 通信では，three way hand shake により，TCP 接続を行い，それからデータ送受信を行い，TCP 通信を開放することで接続を終了する．

- HTTP セッション
  Web サーバと同じネットワーク内にある任意の一つのクライアントとの通信を指す．そのクライアントが Web ページにアクセスしてから，そのページを離れ

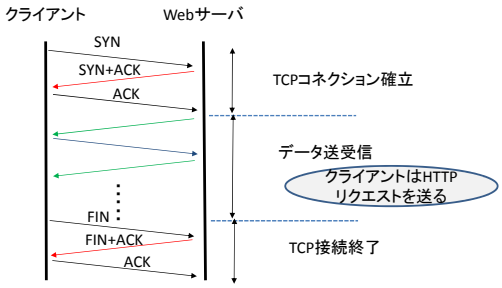


図 3 HTTP セッション

るまでの一連の通信を 1 セッションとする． データは，前述した通り，研究室内に流れるパケットを通常トラフィックとし，ポート番号 80 を利用した通信から HTTP セッションを取り出した．データとしては 2012 年 8 月に採取したデータのうち 1 日分のデータを用いた．

- C&C セッション
  C&C サーバと同じネットワーク内のにある任意の一つのボットとの通信を指す．そのボットが Web ページにアクセスしてから，そのページを離れるまでの一連の通信を 1 セッションとする．
  データは *CCC_DATA_Set*2009, *CCC_DATA_Set*2010 のトラフィックデータの中から C&C セッションを取り出した．ボットが C&C サーバから命令コマンドを受け取る際にはリクエストを送ることが知られている．そこで，パケットのデータの中身に “GET”,“POST”, “HEAD” などのコマンドを送っているトラフィックを抽出した．HTTP 通信は通常ポート番号 80 を利用した通信であるが，C&C セッションではポート番号 80 を利用しているとは限らず，他のポート番号を利用して通信を偽る特性があるため，ポート番号 80 以外でも HTTP リクエストを送っているトラフィックを取り出して，通信の中身をみて，Web コンテンツであれば，それを HTTP 型の C&C セッションとした．

5.3 TCP/IP

TCP/IP とは，インターネット上で用いられる通信プロトコルである．OSI 参照モデルでは，IP がネットワーク層，TCP がトランスポート層にあたり，HTTP や IRC などのネットワーク・プロトコルの基盤となっている．本研究では，TCP/IP プロトコルを用いた通信に焦点を置く．

コンピュータ通信においてデータを送受信する際，データは，IP パケットに分割して送受信される．Web，メールやファイル共有など，どのようなアプリケーションを使う場合でも IP パケットで運んでいる．図 4 を見て分かるように，IP パケットは MAC フレームに格納されることにより ILAN アダプタから LAN ケーブルに送り出される．IP パケットの中身としては，IP ヘッダと IP データに分かれている．IP ヘッダには，パケット長や，フラグ，プロトコル，IP アドレスといった IP パケットの様々な情報が含まれている．今回の実験では，この IP ヘッダ情報を主に用いた実験を行う．

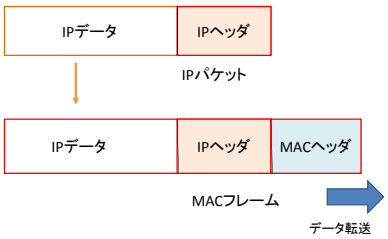


図 4 IP パケットの送信の仕組み

IP ヘッダ情報の中でも，IP アドレスについて述べる．宛先 IP アドレスはその IP パケットの最終目的地を示す．Web を利用している場合，その Web サーバーの IP アドレスが宛先 IP アドレスとなる．送信元 IP アドレスは送信元となるコンピュータに割り当てられた IP アドレスとなる．IP アドレスには「グローバルアドレス」と「プライベートアドレス」がある．図 5 にはプライベートアドレスからグローバルアドレスに変換される仕組みについて示した．インターネットに直接つながっている機器はグローバルアドレスが割り当てられている．それに対して，プライベートアドレスは LAN などに繋がれた端末が自由に使用できる IP アドレスのことである．プライベートアドレスの背景には IPv4 の IP アドレス枯渇問題がある．この問題から，中国やインド，アフリカ諸国では，積極的に IPv6 への移行を進めている．

5.4 特徴ベクトルの定義

通常の HTTP セッションと C&C セッションを分類するために，それぞれの特徴を抽出するための特徴ベクトルを定義した．このとき，既存研究 [1] の要素を用いて定義した．これらの抽出した特徴量は，テキストファイルに出力するなどして，学習データ，テストデータを作成した．

5.4.1 セッション情報

図 6 は，1 つのセッション情報から得られる特徴ベクトルを示す．このベクトルを本研究では，特徴ベクトル A と呼ぶことにする．ここでは 5 つの要素を用いた．具体的な

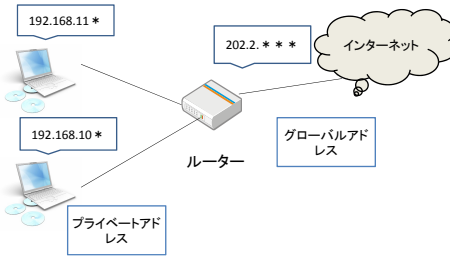


図 5 プライベートアドレスからグローバルアドレスへの変換

特徴の要素としては，

- 送信パケット数
- 受信パケット数
- 送信パケットサイズ
- 受信パケットサイズ
- セッション時間

とした．

5.4.2 パケット列情報

図 7 は，パケット列情報から得られる特徴ベクトルを示す．このベクトルを本研究では，特徴ベクトル B と呼ぶことにする．具体的な特徴の要素としては，

- 送信パケットサイズ
- パケットの送信間隔
- 受信パケットサイズ
- パケットの受信間隔

とした．これらの要素を，1 セッションあたりそれぞれセッションを確立してから n パケット分づつの特徴量を抽出した．本研究では，n=4,8,16 として，実験を行った．

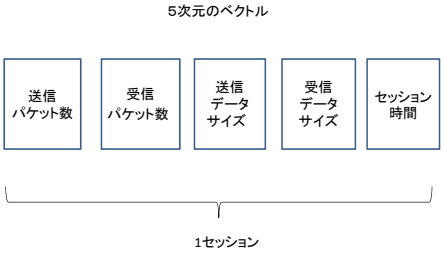


図 6 セッション情報に着目した特徴ベクトル

5.4.3 ヒストグラム情報

図 8 は，データサイズと，パケットの送受信間隔を，ヒストグラム形式にするための，特徴ベクトルを示す．このベクトルを本研究では，特徴ベクトル C と呼ぶ．また，こ

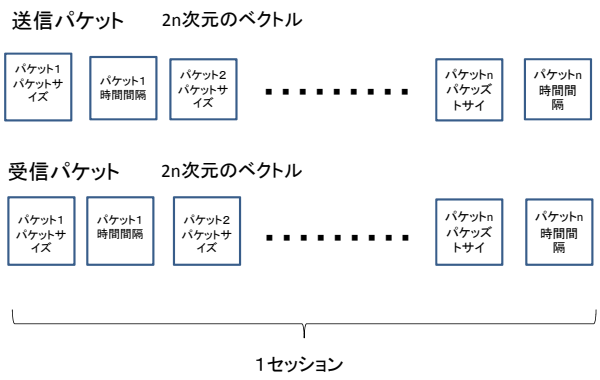


図 7 パケット列情報に着目した特徴ベクトル

の図 7 では，ある 1 つのセッションにおいて，特徴ベクトル C を生成する場合の例を示す．データサイズについては，0-100(byte)，100-200(byte)，200-300(byte)，・・・・・・1500-1600(byte)，のように 16 個の範囲に分割する．パケットの時間間隔については送信パケット，受信パケットは考慮せずに，0-0.1(s)，0.1-0.2(s)，0.2-0.3(s)，・・・・・・0.9-1.0(s)，のように 10 個の範囲に分割する．以上の 2 つの範囲を用いて合計 160 個の範囲に分割し，どのような特徴が現れるかを検証する．

5.4.4 アクセス時間情報

図 9 は，1 セッションの中で，クライアントがサーバにアクセスする時間間隔に関する情報についての特徴ベクトルを示す．このベクトルを本研究では，特徴ベクトル D と呼ぶ．

データサイズ＼時間間隔	1	2	3	4	5	6	7	8	9	10
1	4	1	2	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	0	1
3	2	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	1	1	2	1
5	0	0	0	0	0	0	0	0	0	0
6	3	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	1	2	1	2	1	1	1	1	1	1
9	0	1	1	1	1	1	1	1	1	1
10	1	0	0	0	0	0	0	0	0	0
11	1	2	0	0	0	0	0	0	0	0
12	0	0	0	0	1	1	1	0	0	2
13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	1
15	0	0	0	0	0	0	0	0	0	2
16	0	1	1	1	2	2	1	0	0	0

160次元のベクトルを生成する

(4,1,2,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,2,・・・・・・,2,1,2,1,1,1,1,1,1,・・・・・・,0,1,1,1,2,2,1,0,0,0)

図 8 ヒストグラム形式にした特徴ベクトル

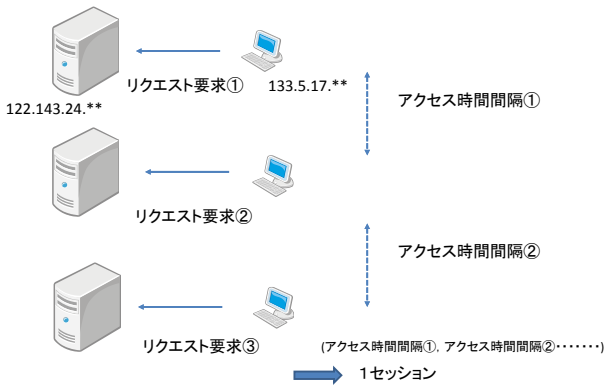


図 9 アクセスの時間間隔に着目した特徴ベクトル

5.5 サポートベクターマシン

それぞれの特徴ベクトルを，サポートベクターマシンを用いて分類した際の精度の評価を行った．サポートベクターマシンは前述したように，パターン認識のアルゴリズム手法であり，学習データが必要となる．そのため，学習データで読み込ませた後，テストデータを用いて分類精度を検証する．また，この実験での分類精度の評価は，検知率，誤検知率，見逃し率の値がどの程度の値となるのかを検証した．

6. 実験評価

6.1 実験結果

表 2 では，それぞれの特徴ベクトルを SVM で分類した結果を示す．また，表 3 では，SVM が学習データで学習してからテストデータを分類するまでの実行時間を示す．これらの考察について，次のセクションで行う．

表 2 実験結果

	検知率 (%)	誤検知率 (%)	見逃し率 (%)
A	0	0	100
B (n=4)	100.0	18.5	0
(n=8)	71.0	51.9	29.0
(n=16)	100.0	45.0	0
C	81.4	1.3	18.6
D	97.5	34.2	2.5

6.2 考察

図 10 は表 2 で得られた実験結果を棒グラフで表したものであり，これをもとに考察をしていく．

- 特徴ベクトル A

表 3 実験結果

	実行時間 (s)
A	19.44
B(n=4)	4.42
(n=8)	0.54
(n=16)	0
C	3.66
D	190.22

特徴ベクトル A に関しては良い分類精度の結果が得られなかった．原因としては，HTTP セッションの中でも，C&C セッションとほぼ一致した特徴量データがあり，分類が困難となってしまったことが挙げられる．

送信パケット数，受信パケット数で見ると，HTTP セッションの方が C&C セッションよりも大きいものもあったが，C&C セッションとほぼ同一な値であった．また，受信データサイズで見ると，HTTP セッションでは受信データサイズが大きいものが多い傾向にあった．それに比べて C&C セッションの受信データサイズは小さく，データサイズの値もほぼ同一であった．さらに，セッション時間について見ると，C&C セッションでは値のばらつきは少ないのに対し，HTTP セッションでのばらつきは大きかった．

結果的に HTTP セッションの特徴量のばらつきが大きいため，C&C セッションも誤って HTTP セッションと判定してまったと考えられる．ボットは通常の HTTP プロトコルを用いるため，トラフィックも通常のものに近くなるため要素数が少ない場合は識別が難しいのでは，と考えられた．

- 特徴ベクトル B

特徴ベクトル B に関しては，n=4,8,16 の全ての場合において誤検知率が高いことが分かった．特に n の値が大きくなる程，精度が悪くなることも分かった．なお，n=16 の場合は学習データから，テストデータを全て C&C セッションとみなしてしまったことから，検知率は高いが誤検知率も高く精度は良くないという結果になった．このことから，2 つのクラスに分類するための特徴は，セッションを確立してから 8 パケット目までにあることが分かった．また，8 パケット目以降に関しては，2 クラスに分類するための特徴は見られなかったということも分かった．

- 特徴ベクトル C

特徴ベクトル C に関しては，誤検知を完全に防ぐことはできなかったが，1.3%と低い値で抑える事ができた．見逃し率が 18.6%と高いが，今回の実験においては比較的良い結果が得られたと言える．

- 特徴ベクトル D
  特徴ベクトル D に関しては，高い検知率を得る事ができたが，誤検知率が 34.2%と高かった．通常の Web サーバとクライアントの通信の場合，同サイト内でページを移動する場合などに Web サーバにリクエストを要求するため，複数回のアクセスが有るものが多かったが，一回しかアクセスしないものもあった．ボットと C&C サーバの通信では，リクエストの要求を 1 回しか行わないものが多かったため，アクセスを 1 回しか行わなかったセッションに関しては，全て C&C セッションとして分類されたと考える．また，今回はトラフィックデータを 20 分毎に観測したため，1 回しかアクセスを行わないものが多かったと考えられる．そのため，トラフィックデータの観測時間を増やすことにより，誤検知は少なくなるのではないかと考えた．

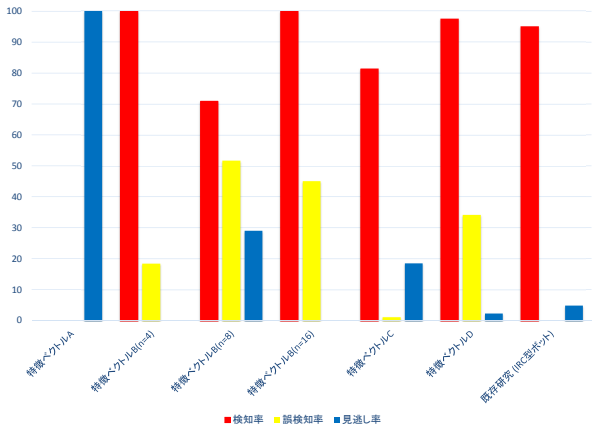


図 10 実験結果

6.3 まとめ

セッションによる情報に焦点を置いた特徴ベクトル A に関しては，あまり良い結果が得られなかった．パケット列の情報に焦点を置いた特徴ベクトル B に関しては，n=4 の時に比較的良い結果が得られた．ヒストグラム形式にした特徴ベクトル C に関しては，比較的良い結果が得られた．アクセス時間間隔に着目した特徴ベクトル D に関しては，誤検知率が高いのが問題点であったが，トラフィックの観測時間を長くすることで，それぞれのセッションにおいてユニークな特性が現れてくるのではないかと考えた．また，精度を上げるためにも，この手法に加えてホワイトリストやブラックリストなどと連携したボットネットの検知も必要であると考えた．

図 10 は，既存研究では IRC 型ボットにおける分類結果を示している．既存研究の結果と本研究での実験結果とを比較すると，本研究での実験結果のほうが，誤検知率，見逃し率は高い．また，データ数に関しても，既存研究のデータ数の方が多いため，信頼性も高い．これらの点を改善すべき点であると考える．

7. 結論

特徴ベクトルを用いた SVM の通常の HTTP セッションと C&C セッションの分類精度に関しては，既存研究に比べると良い結果が出なかったが，特徴ベクトル D に関しては改善点が見つけられた．

8. 今後の課題

本研究で定義した特徴ベクトル D に関して，トラフィックの観測時間を長くした場合の分類精度の評価を行う必要がある．また，C&C セッションのデータ収集を行う際に *CCC_DATA_set* だけでは，データ数が少ないので，データセットの充実化を図りたい．そのために，ハニーポットを構築することによりボットを収集して，ボットの挙動について観察したい．

謝辞 本研究の一部は，JSPS 科研費 23300027，（財）電気通信普及財団の研究助成によって行われた．

参考文献

[1] Satoshi Kondo, Naoshi Sato, “Botnet Traffic Detection Techniques by C&C Session Classification Using SVM”, *IWSEC 2007, LNCS 4752*, pp.91-104, 2007
[2] Daryl Ashley, “AN ALGORITHM FOR HTTP BOT DETECTION” *Daryl Ashley Senior Network Security Analyst University of Texas at Austin - Information Security Office ashley@infosec.utexas.edu*,January 12,2011
[3] 朝長 秀誠, 田中 英彦, “Botnet の命令サーバドメインネームを用いた Bot 感染検出方法” 情報処理学会 *CSS2006*,pp.13-18 2006 年 12 月
[4] Amit Kumar Tyagi, Sadique Nayeem, “Detecting HTTP Botnet using Artificial Immune System”, *International Journal of Applied Information Systems 2(6): Published by Foundation of Computer Science, New York, USA*,34-37, May 2012.
[5] Guofei Gu,Junjie Zhang, and Wenke Lee, “BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic” *School of Computer Science, College of Computing Georgia Institute of Technology Atlanta, GA 30332 guofei, jjzhang, wenke@cc.gatech.edu*
[6] 小堀 智弘, 菊池 浩明 , 寺田 真敏, “マルウェアの通信履歴と定点観測の相関について” 情報処理学会 マルウェア対策研究人材育成ワークショップ *2008* (Oct.8-10, 2008)
[7] PC 遠隔操作の恐怖 `http://sankei.jp.msn.com/affairs/news/121017/crm12101700250001-n1.htm`
[8] IPv4 アドレス在庫枯渇 `http://www.atmarkit.co.jp/ait/articles/1205/14/news149.html`