情報処理学会論文誌 Vol.53 No.9 2062–2074 (Sep. 2012)

トラヒ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

市野 将嗣[1,a)] 市田 達也[2] 畑田 充弘[3] 小松 尚久[2]

受付日 2011年11月30日, 採録日 2012年6月1日

概要：本論文では，トラヒ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を提案する．近年，マルウェアによる被害が多く報告されており，それらの対策として感染検知は不可欠である．そこでマルウェア感染時の通信トラヒックと正常時の通信トラヒックを段階的に識別することで感染の検知を行うシステムを検討する．感染検知をするにあたってトラヒックデータから特徴量を抽出し，それらに対して識別器を用いた判定を行う．本研究では，実用性も考慮して識別アルゴリズムに AdaBoost を用い，AdaBoost の特徴をふまえた時系列データの感染検知手法について検討した．本論文では，研究用データセット CCCDATASet の攻撃通信データを用いた実験結果を報告し，提案手法の有効性を示す．

キーワード：マルウェア，感染検知，AdaBoost，スコア

A Study on Malware Detection Method Based on AdaBoost Using Time Series Traffic Data

Masatsugu Ichino[1,a)] Tatsuya Ichida[2] Mitsuhiro Hatada[3]
Naohisa Komatsu[2]

Received: November 30, 2011, Accepted: June 1, 2012

Abstract: We propose a method of malware detection using time series traffic data. Much damage by malware attack has been viewed recently. We studied the malware detection method by identifying traffic gradually. So we design the classifier to identify malware traffic. We use the AdaBoost as a classification algorithm considering practicability and study a method of malware detection using time series traffic data. In this paper, we evaluated the effectiveness of proposed method by using CCCDATASet.

Keywords: malware, malware detection, AdaBoost, score

1. まえがき

近年のインターネットの普及により，マルウェアの脅威が広がっている．マルウェアとは悪意のあるソフトウェア（Malicious Software）の略称であり，その被害は個人情報の流出やパソコンの乗っ取りというように我々の生活を脅かす存在となっている．マルウェアによる被害は拡大・深刻化しており，近年では活動が表面化しないボットネットによる被害の増加やガンブラ [1] に代表される Web からの感染が増加しているという現状で，早急に対策を講じる必要がある．

マルウェアの検知を侵入検知と感染検知に大別し，本論文では感染検知を目的としている．侵入検知は，マルウェアに感染する前にネットワーク上での不正アクセスを検知する技術である．感染検知は，マルウェアに感染後のトラヒックから感染していることを検知する技術である．近年のマルウェアによる感染は気付きにくい [2] ため，各ユー

[1] 電気通信大学大学院情報理工学研究科総合情報学専攻
Graduate School of Informatics and Engineering, University of Electro-Communications, Chofu, Tokyo 182–8585, Japan
[2] 早稲田大学理工学術院基幹理工学研究科情報理工学専攻
Faculty of Science and Engineering, Waseda University, Shinjuku, Tokyo 169–8555, Japan
[3] NTT コミュニケーションズ株式会社
NTT Communications Corporation, Minato, Tokyo 108–8118, Japan
[a)] ichino@inf.uec.ac.jp

ザが使用しているPCに対して知らない間に感染を拡大させてしまうということが問題となっている．感染の拡大を防ぐためにも各ユーザが使用しているPCやその上流のネットワークにあるルータやファイアウォールなどのネットワーク機器での感染検知は重要な対策である．

本研究では，トラヒックデータを用いた感染検知に着目する．これは，正常時通信トラヒック，感染時通信トラヒックの特徴をとらえて，パターン認識の技術を用いて感染を検知する手法である．トラヒックデータを用いた感染検知は，対象とする機器での通信トラヒックの入出力のみを用いる手法である．基本的には感染すると何らかのトラヒックが発生するため，トラヒックデータを用いた感染検知は対象とする機器の外部からの感染検知手法として有望である．さらに，現在，利用が進んでいる家庭の家電製品や会社の機器端末では必ずしもCPUやメモリが十分ではないため検知モジュールの導入や更新が難しいが，外部からトラヒックのやりとりを見ることでマルウェアの検知を行うことによりそれらのネットワーク接続の際のマルウェア感染検知への適用も期待できる．

通常のブラックリスト方式は，リストの更新時点で既知のマルウェアのみにしか対応できない，変種や亜種などを含む未知のマルウェアが増加しているため，ブラックリストが膨大となり対策が遅れてしまうという問題がある．パターン認識を用いた手法は，感染トラヒックの特徴をとらえることで，ブラックリスト方式では検知できない未知のマルウェアも検知できる可能性がある．また，感染トラヒックの特徴を学習時に登録しておくので，ブラックリスト方式のようなリスト更新の手間が必要ない（登録してある感染トラヒックの特徴と異なるマルウェアが出現した場合には，感染トラヒックの特徴を追加する必要がある．ただし，共通の挙動を持つ未知のマルウェアが多く存在するため，ブラックリスト更新の手間に比べるとパターン認識を用いた手法における，登録してある感染トラヒックの特徴の更新の回数は少ないと考えられ手間は少ない）．つまり，通常のブラックリスト方式はFP，FNを重視し既知のマルウェア感染検知が適用領域であり，パターン認識を用いた手法は，FNをある程度許容しつつ未知のマルウェア感染検知への対応も考慮したものである．近年，未知マルウェアが増加している状況において後者の重要性が高まっている．以上をふまえ，マルウェアの感染検知を行うためにパターン認識の技術に基づく識別器の設計について検討した．

パターン認識の技術に基づく識別器の設計にあたり，本検討のポイントは，トラヒックの時間的な変化を考慮している点である．フロー単位（送受信IPアドレス，送受信ポート番号，プロトコルの5項目が同一のパケット群）で検討する手法の多く [3], [4], [5] は，到着間隔の平均値などを計算することで，フロー全体を1つの特徴量に丸めており，トラヒックの時間的な変動は特徴量に入らない．つまり，連続的な入力が仮定できるにもかかわらず，トラヒックの時間的な変化をひとまとめにして扱っている．このため，トラヒックの時間的な変化に着目することによりさらに識別性能が向上する可能性がある．本研究では，トラヒックをフローではなくスロット単位（トラヒックデータを一定時間間隔で切り出したもの）でとらえ，複数スロットにわたっての振舞いの「変化」を使って，感染検知を行う．

複数スロットにわたっての振舞いの「変化」を使った感染検知の実現のために，トラヒックの時間的な変化を各スロットの「スコアの変化」によって評価する．本研究の提案のポイントは複数スロットによる識別で工夫点を盛り込むことでAdaBoostを適用する方法である．具体的な工夫点とは，「AdaBoostのスコアはベイズ則に一致する」ことを考慮する．AdaBoostのスコアはベイズ則に一致するため正方向に絶対値が大きいほど正常時通信トラヒックらしいということができる．また，負方向に絶対値が大きいほど感染時通信トラヒックらしいということができる．この特徴に着目し，スコアの伸び（各スロットのスコアの和で，複数スロットにおける傾きを表す）を利用する．従来研究で用いられているSVMや決定木は「スコアがベイズ則に一致する」という性質を持たないため，スコア値の大小関係がクラスのもっともらしさを表しておらず，スコアの変化によって振舞いの変化を正しく評価できない．

そこで本論文では，トラヒックデータの連続入力を考慮した，AdaBoostに基づくマルウェア感染検知手法を提案し，研究用データセットCCCDATASetを用いた実験で有効性を示す．以下，2章では，マルウェア感染検知手法の従来技術を述べ，3章では，従来技術の問題点を述べる．4章では，連続するデータを考慮したマルウェア感染検知手法を提案する．5章では，研究用データセットCCCDATASetの攻撃通信データを用いた実験結果を示す．6章では，実験結果に対する考察を述べる．7章はまとめと今後の課題である．

2. 従来技術

本章では，従来のマルウェア感染検知を行うための識別器の構成（特徴量，識別アルゴリズム）について説明する．そして従来技術の問題点について述べる．

2.1 識別器

2.1.1 特徴量

特徴量に関しては通信時のトラヒックに着目する．フローではなく，トラヒックをタイムスロットごとに抽出し，スロット内のパケットのヘッダ情報から得られる統計情報に着目し，トラヒックから得られる連続入力データを考慮した感染検知手法を検討した．

フロー単位で検討する手法の多く [3], [4], [5] は，到着間

隔の平均値などを計算することで，フロー全体を 1 つの特徴量に丸めており，トラヒックの時間的な変動は特徴量に入らない．つまり，連続的な入力が仮定できるにもかかわらず，トラヒックの時間的な変化をひとまとめにして扱っている．このため，トラヒックの時間的な変化に着目することによりさらに識別性能が向上する可能性がある．フロー単位で検討する手法は時間的な変動を考慮しないマルウェア感染検知が適応領域であり，タイムスロット単位で検討する手法は時間的な変動も考慮したものである．時間的な変化に着目する場合には後者が必要となる．

トラヒックのタイムスロットから感染を検知するにはペイロード情報を用いる方法 [6], [7], [8] とヘッダ情報を用いる方法 [9], [10], [11] がある．ペイロード情報を用いる方法はプライバシが問題にならず，暗号化されていない通信の感染検知が適用領域である．ヘッダ情報を用いる方法はプライバシの問題[*1]を考慮し，暗号化された通信への適用も考慮したものである．負荷とプライバシの観点からの問題があるので，後者の重要性が高まりつつある．

以上より，本研究では特徴量としてパケットのヘッダ情報から得られる統計情報を用いることとし，トラヒックの時間的な変化を考慮するためスロットごとのパケットのヘッダ情報から得られる統計情報に着目した．

2.1.2 識別アルゴリズム

識別アルゴリズムとしては，ナイーブベイズや Support Vector Machine（SVM），決定木がよく用いられている [9], [10], [11]．これらの研究は，単一のスロットごとに感染トラヒックかどうかを判定している．

また，本研究は感染検知（感染後のトラヒックから感染していることを判定する）を対象としているが，IDS（Intrusion Detection System）において本検討と共通点がある技術である異常値検知で AdaBoost が使われている [12], [13], [14], [15], [16]．Li らは，基本的にはナイーブベイズを AdaBoost で Boost している [12]．Hu らは，各特徴量を単一のタイムスロットごとに AdaBoost を用いて識別している [13]．Zhang らは，基本的には HMM を AdaBoost で Boost している [14]．これらの研究は，単一のスロットごとに感染かどうかを判定している．

また，AdaBoost 以外に複数の弱識別器を組み合わせた異常トラヒック検出手法について提案されている [17]．特徴空間をクラスタの密度を考慮し k-平均法でクラスタリングし，各クラスタ中心との距離を求める弱識別器を作成する．そして単一のスロットごとに感染かどうかを判定している．

2.2 従来技術の問題点

2.1.2 項で述べた研究はいずれも単一のスロットでの感染検知を対象としておりトラヒックの時間的な変化をとらえたものではない．感染時通信トラヒックの中にも正常時通信トラヒックに近いスロットがあるため，単一のスロットによる識別では誤識別が起こるという問題がある．AdaBoost を用いた研究に着目すると，いずれの研究も弱識別器を Boost することに主眼が置かれ，AdaBoost の重要な性質の 1 つである AdaBoost のスコアはベイズ則に一致するという特徴を利用していない．それに対して本提案技術は，AdaBoost のスコアはベイズ則に一致するという特徴を利用しトラヒックの時間的な変化を考慮することにより，さらに TPR，TNR が向上する可能性がある．

ナイーブベイズは単一スロットによる識別であるという問題点以外に，あらかじめ事前確率が分かっていることが前提となっているが，未知マルウェアの事前確率を求めるのは困難である．さらに，入力変数の各成分（パケットのヘッダ情報から得られる統計情報）の分布が独立であるという仮定もあるが，各成分には相関があるため，仮定を満たしておらず識別精度が低下する可能性がある．

また，実用を考慮するとセキュリティレベルにより閾値制御できることが望ましい[*2]．

基本的には SVM，決定木はサンプルどうしの分離の良さを評価関数としているため，必ずしもスコアの値の大小関係がクラスのもっともらしさを表していない．そのため適切な閾値制御ができないという問題がある．また，SVM は識別率が良くなるように特徴ベクトルを構成する特徴量（本論文では，識別に用いるヘッダ情報の統計量を特徴量とする）をあらかじめ決めておく必要がある．SVM，決定木は以上のような実装するうえでの問題点がある．

また，AdaBoost 以外に複数の弱識別器を組み合わせた異常トラヒック検出手法 [17] は，弱識別器をいくつ用意すればよいのか，つまり弱識別器と精度の関係が不明確と考えられる．

3. 提案技術

提案のポイントは複数スロットによる識別で工夫点を盛り込むことで AdaBoost を適用する方法である．具体的な工夫点とは，AdaBoost のスコアはベイズ則に一致することを考慮しスロットごとのスコアの伸び（対象区間内での各スロットのスコアの和で，対象区間における傾きを表す）を利用することである．感染時通信トラヒックの中にも正常時通信トラヒックに近いスロットがあるため，単一のスロットによる識別では誤識別が起こるという問題がある．複数スロットを使うことにより全体のトラヒックで識別で

[*1] 厳密な解釈を選んだ場合，ポート番号を監視することも通信の秘密に触れると指摘される場合もある．

[*2] たとえばネットワークのセキュリティの場合を考えてみると，個人使用の PC 利用時では，FPR（正常のデータを識別器が感染と判定した割合）を下げたいという要件がある．それに対して，重要な情報が保管されている役所などの PC やサーバでは，FNR（感染のデータを識別器が正常と判定した割合）を下げたいという要件がある．このように場面に応じた制御が必要である．

きる可能性がある．

以下，3.1 節では，AdaBoost に着目した理由および本検討での AdaBoost の適用方法を説明する．3.2 節では提案のポイントとなる複数スロットによる識別方法について説明する．

3.1 AdaBoost

上記でも述べたように異常値検出でこれまでに AdaBoost [19], [20] が適用されている [12], [13], [14], [15], [16]．AdaBoost は，適応的にサンプルの重みを更新することにより，単純で弱い識別器（弱識別器）を逐次的に学習し，識別器の精度を増強していく手法である．AdaBoost を用いることで正常時通信トラヒックとマルウェア感染時トラヒックを段階的に識別していくことができる．次の 4 点より AdaBoost はマルウェア感染検知に有効であると考える．

- トラヒックは様々な挙動の組合せであることより特徴空間においては複雑な分布となり非線形もしくは区分線形の識別アルゴリズムが必要であると考えられる．AdaBoost は非線形もしくは区分線形の識別面が作成可能である．
- パケットのヘッダから得られる情報は多くある（文献 [18] では 36 個の特徴量がある）ため識別にどの特徴量を使用すればよいか選ぶのが難しい．AdaBoost には特徴量選択とともに特徴量に重みづけを行うことができるという特徴がある．
- AdaBoost には，弱識別器の数を増やせば増やすほど識別精度が向上する一方，識別時間は増えていくという特徴がある[*3]．つまり識別時間と識別精度を調整できるという特徴がある．識別精度を重視する場合には弱識別器の数を増やし，識別時間を重視する場合には弱識別器の数を減らせばよい．実用向けの特徴である．
- 単純に特徴量の統合により精度向上を狙うのではなく実用を考慮するとセキュリティレベルに応じた閾値の制御によって FPR，FNR を制御できることが望ましい．AdaBoost のスコアはベイズ則[*4]に一致するため閾値制御の根拠を論理的に説明することができる．

AdaBoost は各サンプルの重みを更新することにより，それぞれの重み分布 D_b の下で弱識別器を学習する．具体的には，初期の重みは均等に割り振っておき，その後は弱識別器が正しく識別できたサンプルについては重みを小さくし間違えたサンプルについては重みを大きくする．これにより直前の弱識別器が苦手とするサンプル分布の下で，次の弱識別器が学習される．これは，トラヒックの様々な挙動を表すトラヒックデータから正常と感染を識別するための弱識別器を多数生成していることになる．

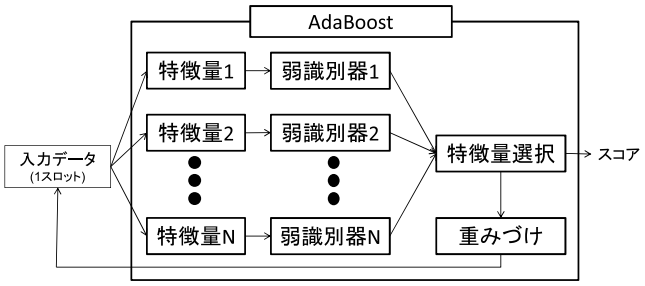

図 1 AdaBoost の概要
Fig. 1 Outline of AdaBoost.

図 1 に AdaBoost を適用する際の概要を示す．一般的な AdaBoost の構成では，識別に用いるすべての特徴量を並べたベクトル（使用する特徴量の数の次元数となる）に対して複数の弱識別器で識別を行うが，図 1 の AdaBoost では，AdaBoost で特徴量選択を行うために b 番目の弱識別器 h_b に全サンプルの b 番目の特徴量のみを入力する．学習・識別は具体的には次のようになる．

【学習フェーズ】

step.1 重みの初期化

各サンプル[*5]の重みを $D_1(i) = \frac{1}{M}$ で初期化する．ただし M は学習データの全サンプル数である．

step.2 弱識別器の学習

(1) $b = 1, 2, \cdots, B$ に対して以下の (2) から (6) の処理を行う．この B は識別の際に用いる弱識別器の数に相当する．

(2) b に対して $j = 1$ から $j = N$ の特徴量に対して以下の識別を行う．

学習データ $t = 1$ から $t = M$ それぞれのスロットに対して，t 番目のヘッダ情報の統計量 x_t を用いて識別を行う．サンプルの重み分布 D_j に基づき各特徴量の j 番目の弱識別器 h_j と各特徴量（弱識別器）の重み（信頼度 α_j）を求め，学習サンプルに対する誤り率

$$\epsilon_j = \Sigma_{i:y_i \neq h_j(x_t)} D_j(i) \tag{1}$$

を計算する．弱識別器を以下のように設計する．

$$h_j(x_t) = \begin{cases} +1 & \text{if} \quad p \cdot FV_j(x_t) > p \cdot \theta \\ -1 & \text{otherwise} \end{cases} \tag{2}$$

ただし，y_i は正解クラス，$FV_j(x_t)$ は t 番目のスロットのヘッダ情報の統計量 x_t における j 番目の特徴量の値，p は特徴量と閾値 θ を比較する不等号の向きを決定する変数で，$+1$ もしくは -1 の値をとる．これは特徴空間上で正常時通信トラヒックのサンプルと感染時通信トラヒックのサンプルの位置関係で不等号の向きを変えられるようにするためである．つまり，$p = 1$ とした場合，$FV_j(x_t) > \theta$ のときに $h_j(x_t) = +1$，それ以外のときに $h_j(x_t) = -1$ と

[*3] 指数損失を単調に減少させることができる．
[*4] 入力サンプルに対して事後確率が最大となるクラスを出力結果とする識別法．
[*5] 本論文では，1 スロットを 1 サンプルとする．

なる．また，$p = -1$ とした場合，$FV_j(x_t) < \theta$ のときに $h_j(x_t) = +1$，それ以外のときに $h_j(x_t) = -1$ となる．θ，p は学習サンプルに対する識別誤りが最小となるように値を求める．

(3) ϵ_j（$1 \leq j \leq N$）のうち，最小となる ϵ_j を ϵ_b とし，対応する特徴量（弱識別器 $h_b = h_j$）を選択する．

(4) 誤り率（式 (1) の ϵ_j）が最小となる弱識別器 $h_b(x_t)$ を選択したときの誤り率 ϵ_b から α_b を

$$\alpha_b = \frac{1}{2}\log\left(\frac{1-\epsilon_b}{\epsilon_b}\right) \tag{3}$$

によって計算する．

(5) サンプルの重みを (4) で計算した α_b を用いて更新する．

$$D_{b+1}(i) = D_b(i)\exp[-\alpha_t y_i h_b(x_t)] \tag{4}$$

ただし，正しく識別できたサンプルについては $D_{b+1}(i) = D_b(i)\exp(-\alpha_t)$，間違えたサンプルについては $D_{b+1}(i) = D_b(i)\exp(\alpha_t)$ とする．

(6) サンプルの重みの和が 1 になるように正規化する．

step.3　強識別器

最終的に得られる強識別器 $H(x_t)$ は，B を弱識別器の数，α_b を弱識別器の重みとすると次式で表される．

$$H(x_t) = sign[\Sigma_{b=1}^{B}\alpha_b h_b(x_t)] \tag{5}$$

【識別フェーズ】

$H(x_t)$ はベイズ則と一致するため，t 番目のスロットの AdaBoost のスコアを

$$s(x_t) = \Sigma_{b=1}^{B}\alpha_b h_b(x_t) \tag{6}$$

とする．$h_b(x_t)$，α_b については学習フェーズで求めたものを使用する．

3.2　複数スロットによる識別

複数スロットを使うことでさらなる精度向上を実現する方法について検討した．

3.1 節で述べたように，AdaBoost のスコアはベイズ則に一致するため正方向に絶対値が大きいほど正常時通信トラヒックらしいということができる．また，負方向に絶対値が大きいほど感染時通信トラヒックらしいということができる．そのことを利用して本研究では，Adaboost の特性と連続的にトラヒックデータを取得できることを考慮し，連続する複数スロットのスコアの伸び（対象区間あたりのスコアの増加もしくは減少の傾きの具合）に着目する．時間的な変化をスコアの伸びで表現する．つまり，スコアの伸びが正方向に大きいほど正常時通信トラヒックらしく，また，スコアの伸びが負方向に大きいほど感染時通信トラヒックらしいということができる．複数スロットによる識別の概要を**図 2** に示す．

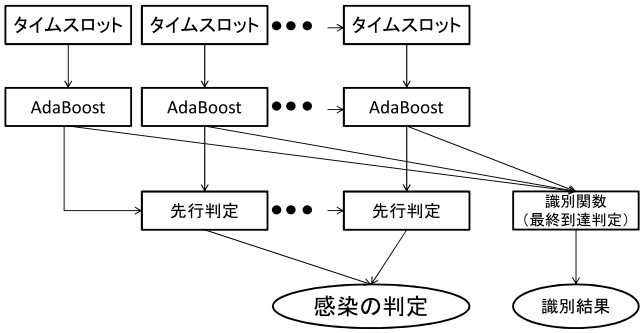

図 2　複数スロットによる識別の概要
Fig. 2　Outline of identification using multi slot.

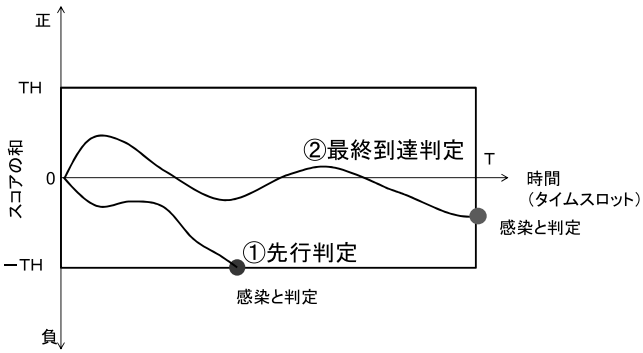

図 3　複数スロットによる識別のイメージ
Fig. 3　Concept of identification using multi slot.

本論文での連続するスロットによるスコアの伸びを利用した識別のイメージを**図 3** に示す．スコアの統合に着目し，複数スロットより求まるスコアを統合した形で識別することを考える．本論文では，AdaBoost によるスコアはベイズ則に一致することをふまえ，スコアの伸び（対象区間内での各スロットのスコアの和で，対象区間における傾きを表す）を

$$S(x) = \Sigma_{t=1}^{T}s(x_t) \tag{7}$$

と表し，識別関数を

$$I(x) = sign[\Sigma_{t=1}^{T}s(x_t)] \tag{8}$$

のように表し，対象区間（区間幅を T とする）におけるスコアの伸びを全体スコアとして扱う（図 3 の ② に相当し，以下，本論文では「最終到達判定」と呼ぶ）．これは「全体のトラヒックを見て厳密に識別する」ことを意味する．また，スコアの和が閾値 $-TH$ に達した段階でスコアの伸びが大きいとして感染と判定することとした（図 3 の ① に相当し，以下，本論文では「先行判定」と呼ぶ）．つまり，1 スロット経過するたびにスコアの伸びを計算し，スコアの伸びを用いて感染を判定する．この方法は，「明らかにマルウェアに感染しているものから識別していく」と定性的に説明できる．

本論文ではスコアの伸びを利用した「先行判定」と「最終到達判定」を組み合わせた検知を提案する．「先行判定」を

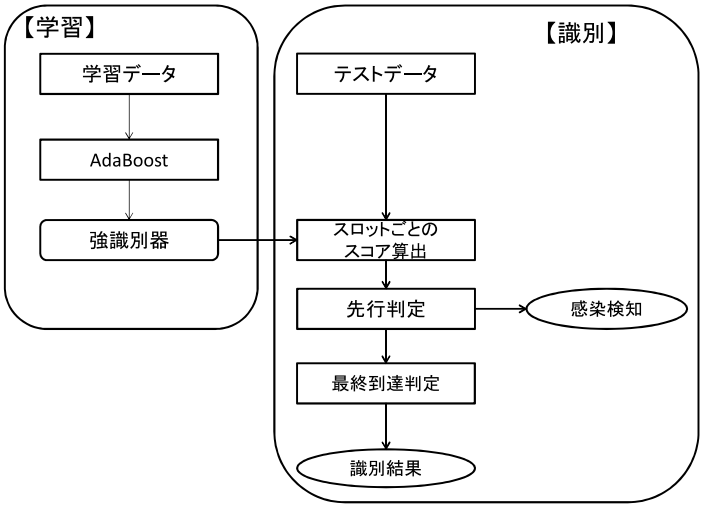

図 4 実験系の概要
Fig. 4 Experimental system.

行い明らかに感染であるトラヒックを先に検知する．「先行判定」で検知できていないものを「最終到達判定」を行うことで感染トラヒックを時間をかけてより高精度に検知する．

4. 評価実験

本章では，識別アルゴリズムに AdaBoost を用い，トラヒックデータの連続入力に着目した感染検知手法の実験結果を示す．

4.1 実験系の概要

実験系の概要を図 4 に示す．

学習では，正常時通信データ，マルウェア感染時通信データそれぞれからタイムスロットごとに特徴量（ヘッダ情報から得られる統計情報）を求め[*6]，AdaBoost を適用し，強識別器を作成した．

識別では，学習と同様にテストデータからタイムスロットごとに特徴量を求めた．そして，各スロットごとに学習で作成した強識別器を用いてスコアを求めた．先行判定では，各スロットのスコア $s(x_t)$ の和が $-TH$ を下回ったときに感染と判定した．最終到達判定では，0 から T までのスロットのスコア $s(x_t)$ の和（全体スコア）が正であれば正常，負であればマルウェアに感染していると判定した．

4.2 実験データ

実験データとして，正常時通信データと感染時通信データを用意する必要がある．正常時通信トラヒックデータとしては，あるイントラネットより取得したデータ，一方感染時通信トラヒックデータとして CCC2010，2011 [21] を用いた．今回の評価実験では，正常トラヒックデータと感染トラヒックデータは取得環境が異なっている．

学習については，特徴空間上で正常トラヒックと感染トラヒックを識別するための識別器を作成するために，正常トラヒック，感染トラヒックそれぞれからタイムスロットごとにヘッダ情報の統計情報を抽出した後，それらのサンプル（タイムスロットごとのヘッダ情報の統計情報）を特徴空間上に散布させた．識別時においても，識別対象のトラヒックが正常トラヒックか感染トラヒックかを AdaBoost で作成した強識別器で識別することとなるので，学習データとテストデータには正常トラヒックと感染トラヒックは混ぜていない．

正常時トラヒックデータは 2010 年 3 月 7, 8 日と 2011 年 1 月 21，22 日のデータ，感染時トラヒックデータは 2010 年 3 月 5 日から 10 日と，2011 年 1 月 18 日から 23 日までのデータを使用した．また，感染時データに関しては CCC2010，2011 内のログ情報をもとに明らかにマルウェアに感染していると考えられる通信データを切り出して用いた．サイバー空間での攻撃の傾向は日々変化しているため，ある年のデータセットに最適であった手法が別の年のデータセットに普遍的に適用できるとはかぎらない．そこで，学習データとテストデータの組合せについて

(1) 学習データとテストデータの取得が同時期の組合せ（学習データ：2010 年，テストデータ：2010 年，もしくは学習データ：2011 年，テストデータ：2011 年）学習データ：22,248 スロット（正常トラヒック：15,920 スロット，感染トラヒック：6,328 スロット）テストデータ：44,843 スロット（正常トラヒック：32,786 スロット，感染トラヒック：12,057 スロット）
(2) 学習データとテストデータの取得が異時期の組合せ（学習データ：2010 年，テストデータ：2011 年）学習データ：13,891 スロット（正常トラヒック：8,517 スロット，感染トラヒック：5,374 スロット）テストデー

[*6] 1 スロットが特徴空間における 1 点に相当する．

タ：36,566 スロット（正常トラヒック：31,694 スロット，感染トラヒック：4,872 スロット）

の 2 パターンの実験を行った．学習用データと評価用データを二分した方法として，同時期での評価についはトラヒックを取得した日付（正常トラヒックは 1 日分を学習用，学習用と同じ年の 1 日分を評価用，感染トラヒックは連続する 3 日分を学習用，学習用と同じ年の連続する 3 日分を評価用）で分けた．異時期の評価についてはトラヒックを取得した年（正常トラヒックは 2010 年の 1 日分を学習用，2011 年の 1 日分を評価用，感染トラヒックは 2010 年の連続する 3 日分を学習用，2011 年の連続する 3 日分を評価用）で分けた．感染トラヒックは正常トラヒックに比べスロット数が少ないため 3 日分を用いた．複数スロットの作り方については学習用データ，評価用データに二分し感染トラヒックは連続する 3 日分を連結した後，連続するスロットから T スロット分をひとまとめとして複数スロットとし，長さ T の複数スロットを作成した．この作業を繰り返し，複数スロットを作成した．

今回使用したトラヒックデータの特性として，イントラネットの特性についは，ホスト数が 25，主な通信プロトコルは HTTP，HTTPS，SMB，SNMP などとなる．トラヒック量は約 1〜150 パケット/秒となる（時間帯により異なる）．CCC データセットの特性についは，ホスト数が 2，主な通信プロトコルは HTTP，DNS，SMB など，トラヒック量は約 0.1〜250 パケット/秒となる．

タイムスロット幅に関して，今回 1 スロットを 10 秒間とし，$T = 12$（2 分に相当，図 3 参照）とした．スロット幅は文献 [18] の検討などに基づき決めた．有効なスロット幅は各特徴量で一致するのではなく，特徴量ごとに，そして TPR，TNR ごとに有効なスロット幅は異なる．特徴量全体を通して，経年変化（たとえば，2009 年に取得したトラヒックデータで学習を行い，2009，2010，2011 年のそれぞれのトラヒックで識別を行った際の識別率の変化）による影響が少なく，TPR，TNR の両方が良くなるスロット幅を採用した．また，スロット幅が小さいと細かいトラヒックの変動を見ることができるが，その分安定した特徴抽出ができない．それに対してスロット幅を少し長めにとるとスロット内の統計量を特徴量とするのでスロット幅が小さいときに比べて安定した特徴抽出を行うことができる．そのため，本検討ではスロット幅を 10 秒とした．また，本研究では，パケットのヘッダ情報から取得できる情報およびその統計値を特徴量として用いる．そこで本実験では，文献 [18] で示されている 36 個の特徴量を用いた．特徴量 36 種類を**表 1** に示す．本論文での特徴量は表 1 に示す 36 種類のヘッダ情報の統計量である．表 1 は CCC データセットから正常トラヒックと感染トラヒックを分離できそうな特徴量の候補を考えたものである．

表 1 特徴量 36 種類
Table 1 36 types of feature.

番号	特徴量 [単位]
1	パケット数
2	パケットサイズの総数 [byte]
3	パケットサイズの平均 [byte]
4	パケットサイズの最小 [byte]
5	パケットサイズの最大 [byte]
6	パケットサイズの標準偏差 [byte]
7	到着間隔の平均 [秒]
8	到着間隔の最小 [秒]
9	到着間隔の最大 [秒]
10	到着間隔の標準偏差 [秒]
11	SYN パケット数
12	FIN パケット数
13	PSH パケット数
14	ACK パケット数
15	RST パケット数
16	URG パケット数
17	SYN/ACK パケット数
18	FIN/ACK パケット数
19	PSH/ACK パケット数
20	RST/ACK パケット数
21	TCP パケット中の SYN パケット割合
22	TCP パケット中の FIN パケット割合
23	TCP パケット中の PSH パケット割合
24	TCP パケット中の ACK パケット割合
25	TCP パケット中の RST パケット割合
26	TCP パケット中の URG パケット割合
27	TCP パケット中の SYN/ACK パケット割合
28	TCP パケット中の FIN/ACK パケット割合
29	TCP パケット中の PSH/ACK パケット割合
30	TCP パケット中の RST/ACK パケット割合
31	ICMP 到達不能メッセージ数
32	UDP パケット数
33	送信元ポート番号が 69/UDP のパケット数
34	送信元ポート番号が 80/TCP のパケット数
35	送信元ポート番号が 110/TCP のパケット数
36	送信元ポート番号が 443/TCP のパケット数

4.3 実験結果

本論文での提案は「先行判定」と「最終到達判定」を組み合わせたものである．マルウェアの感染検知における「先行判定」の位置づけは，明らかに感染であるものを先に検知するということである．一方，「最終到達判定」の位置づけは，感染トラヒックを時間をかけてより高精度に検知するということである．そこで本論文ではそれぞれの効果を見るために Best なパフォーマンスが出るパラメータ（T，TH）の値に設定し，それぞれの効果を評価した．

AdaBoost を用いることの有効性を確認するために SVM，決定木（C4.5），ナイーブベイズとの比較を行った．さらに，「先行判定+最終到達判定」による識別の有効性を確認するために単一スロットによる識別と比較を行った．次に

表 2 識別結果
Table 2 Identification results.

識別手法	同時期			異時期		
	識別率	TPR	TNR	識別率	TPR	TNR
SVM	91.0%	67.7%	99.6%	92.2%	41.8%	99.9%
決定木（C4.5）	99.8%	99.5%	99.9%	89.8%	66.2%	99.7%
ナイーブベイズ	73.4%	69.4%	84.3%	60.4%	59.3%	67.5%
AdaBoost（先行判定+最終到達判定）	99.8%	99.3%	100.0%	100.0%	100.0%	100.0%
AdaBoost（単一スロット）	98.6%	95.2%	99.9%	92.9%	91.9%	99.4%

先行判定と最終到達判定それぞれの効果を見るために，単独で用いたときの効果を評価した．

単一スロットによる識別とは，式 (6) のスコアの正負により，1 スロットごとに正常・感染の識別を行う方法である．AdaBoost については，$B = 10$ とした．SVM，決定木，ナイーブベイズについては，特徴量を 36 個使用した．SVM 適用の際のカーネル関数はガウス型動径基底関数

$$k(\vec{x}, \vec{y}) = \exp\left(\frac{-||\vec{x} - \vec{y}||^2}{2\sigma^2}\right) \tag{9}$$

を用い，予備実験により適切と思われる σ を設定し，SVM^{light} [22] を使用して識別を行った．決定木，ナイーブベイズについては WEKA [23] を使用した．学習データとテストデータの取得が同時期，異時期の組合せの識別結果[*7]を**表 2** に示す．表 2 に，単一スロットによる識別の結果（AdaBoost（単一スロット））と 3.2 節で示した先行判定+最終到達判定による識別の結果（AdaBoost（先行判定+最終到達判定））もあわせて示す．AdaBoost（先行判定+最終到達判定）の最終到達判定は 0 から T（今回は $T = 12$）までのスロットのスコアの和 $S(x)$（式 (7)）が正であれば正常，負であればマルウェアに感染していると判定した結果である．

表 2 より，AdaBoost の 2 手法は SVM，決定木，ナイーブベイズに比べ同時期，異時期において TPR，TNR が安定しており，高い識別率が得られていることが分かる．SVM は $\sigma = 5.0$ で高い識別率を示した．SVM は特徴量選択が行われていないため正常データと感染データの分布が複雑に重なっていると考えられ，そのため複雑な識別面が必要となり σ の値が小さいと考えられる．

また，表 2 より先行判定+最終到達判定による識別は単一スロットによる識別に比べ，高い識別性能が得られていることが分かる．

最終到達判定による識別（3.2 節）では T スロットのスコアの和を全体スコアとし，その全体スコアが正であるか負であるかで識別を行う．T の値により識別性能が変わるため，T と識別性能の関係を調べた．T の値と識別率の関係を**図 5** に示す．図 5 より T を大きくすると識別率が上がることが分かる．

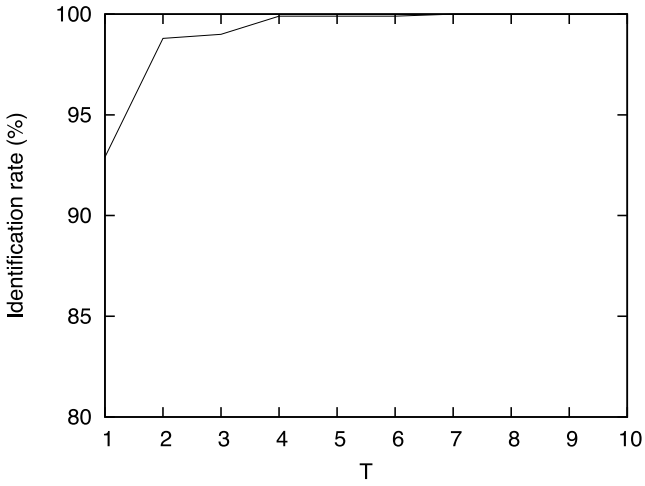

図 5 T と識別率の関係
Fig. 5 T and identification rate.

先行判定では，1 スロット経過するたびにスコアの伸びを計算し，スコアの伸びを用いて感染を判定する．今回の実験では，先行判定による識別は，スコアの和が TH に達するかどうかで判定する．TH の値により識別性能が変わるため，TH と識別性能の関係を調べた．先行判定の際に使用する閾値 TH（図 3 の TH）に対する感染検知率[*8]（TPR）および使用した平均スロット数の関係を**図 6** に示す．実線が TPR，破線が使用する平均スロット数となる．また，先行判定する閾値 TH に対する誤検知率[*9]（FNR）および使用した平均スロット数の関係を**図 7** に示す．実線が FNR，破線が使用する平均スロット数となる．

図 6 に着目すると，$TH = 3$ のときに，TPR=100%となる．このときの平均スロット数が 2.4 となるので，最終到達判定に比べ早く感染を検知できることが分かる．

閾値 TH を大きくするにつれて．感染検知率，誤検知率

[*7] 識別率：テストデータを正しく識別（正常，感染）した割合，TPR：感染時通信のテストデータを識別器が感染と判定した割合，TNR：正常時通信のテストデータを識別器が正常と判定した割合．

[*8] 感染データを感染していると検知する割合．

[*9] 感染データを正常（スコアの和が $+TH$ より大きい）と判定した割合．

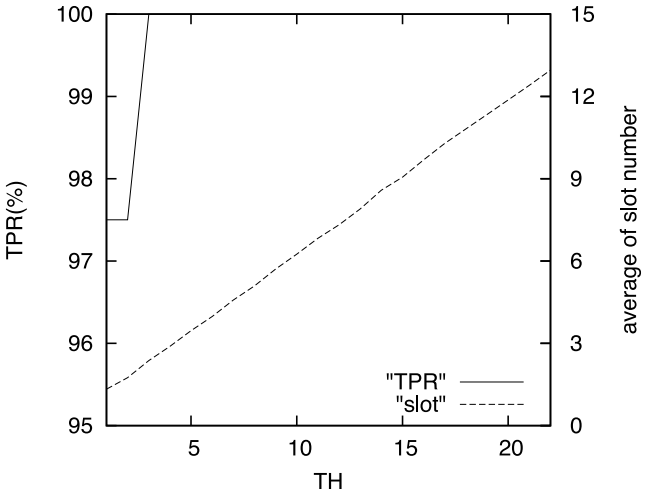

図 6 閾値 TH に対する感染検知率と平均スロット数の関係
Fig. 6 TPR and average slot for TH.

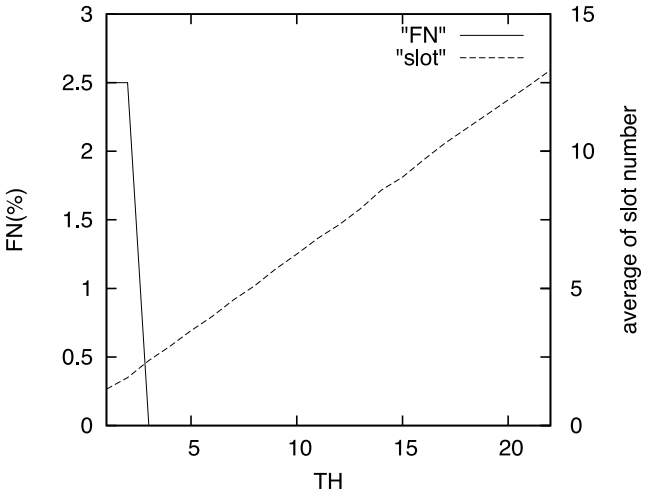

図 7 閾値 TH に対する誤検知率 (FNR) と平均スロット数の関係
Fig. 7 FNR and average slot for TH.

が良くなっていることが分かる．ただし，TH に到達するのに必要なスロット数が多くなるので，感染を検知するまでの時間が長くなる．

5. 考察

5.1 AdaBoost により選択された特徴量

今回の実験において，AdaBoost により選択された特徴量は，UDP パケットの数，パケットサイズの最小，パケットサイズの最大，到着間隔の最小，送信元ポート番号が 443/TCP のパケット数，TCP パケット中の SYN パケット割合などである．AdaBoost ではこれらの特徴量に重みをつけ，使用する順番を変えて利用している．$B = 10$ のときの Boosting により選択された特徴量を**表 3** に示す．

文献 [18] では，マルウェア感染検知の既存研究でよく用いられている特徴量に対して，特徴量ごとにベクトル量子化で作成した正常時，感染時のコードブックとテストデータとの特徴空間上での距離を用いて識別を行い，マルウェアに感染している感染トラヒックとマルウェアに感染していない正常トラヒックの識別実験により特徴量評価を行った結果が報告されている．その実験において，今回選択された特徴量は，文献 [18] の評価においても TPR，TNR が高い特徴量として選ばれており，特徴量選択も有効に機能していると考えられる．

表 3 Boosting により選択された特徴量
Table 3 feature selected by boosting.

Boosting 回数	Boosting により選択された特徴量
1	UDP パケット数
2	パケットサイズの最小
3	パケットサイズの最小
4	パケットサイズの最大
5	到着間隔の最小
6	送信元ポート番号が 443/TCP のパケット数
7	TCP パケット中の SYN パケット割合
8	パケットサイズの最小
9	パケットサイズの最小
10	パケットサイズの最小

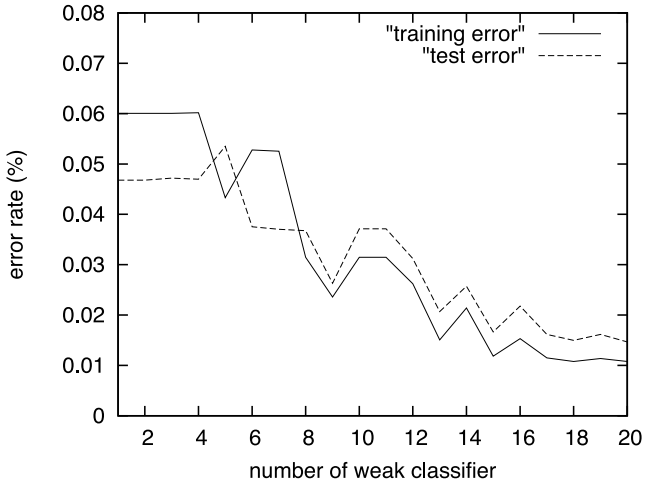

図 8 訓練誤差と汎化誤差
Fig. 8 Training error and test error.

5.2 Boosting について

本研究では，図 1 に示すように，AdaBoost で特徴量選択を行うために b 番目の弱識別器 h_b に全サンプルの b 番目の特徴量のみを入力する．識別に用いるすべての特徴量を並べたベクトル（使用する特徴量の数の次元数となる）に対して複数の弱識別器で識別を行う一般的な AdaBoost の構成とは少し異なるため，必ずしも Boosting が保証されなくなる．

そこで，Boosting の保証を確認するために，訓練誤差[*10]と汎化誤差[*11]を調べた．訓練誤差・汎化誤差と弱識別器の数の関係を**図 8** に示す．横軸が使用した弱識別器の数，縦軸

[*10] 学習サンプルに対して識別を行った際の誤り率．
[*11] 未学習サンプルに対して識別を行った際の誤り率．

が誤り率を表し，訓練誤差を「training error」，汎化誤差を「test error」で示した．図 8 より，弱識別器の数を増やせば増やすほど誤り率が低下していることが分かる．つまり識別に使用する弱識別器の数を調整することにより識別時間と識別精度を調整できることが分かる．

5.3 セキュリティレベルに応じた閾値の制御について

AdaBoost のスコアはベイズ則に一致するため，式 (5) に式 (10) のように th を導入し，th を制御することで，セキュリティレベルに応じた閾値の制御を行うことができると考えられる．

$$H(x_t) = sign[\Sigma_{b=1}^{B} \alpha_b h_b(x_t) + th] \tag{10}$$

そのことを確認するために，AdaBoost において式 (10) の th を閾値とした場合の FPR，FNR の関係を調べた．**図 9** に結果を示す．また，実験を行った際に求まったスコアのヒストグラムを**図 10** に示す．実線が正常トラヒック，破線が感染トラヒックを表し，横軸がスコアの値，縦軸がスコアの頻度の割合（スコアの頻度回数を全体の頻度回数で割ったもの）を表す．そして式 (10) の th の制御は図 10 の横軸上を動かすことに相当する．

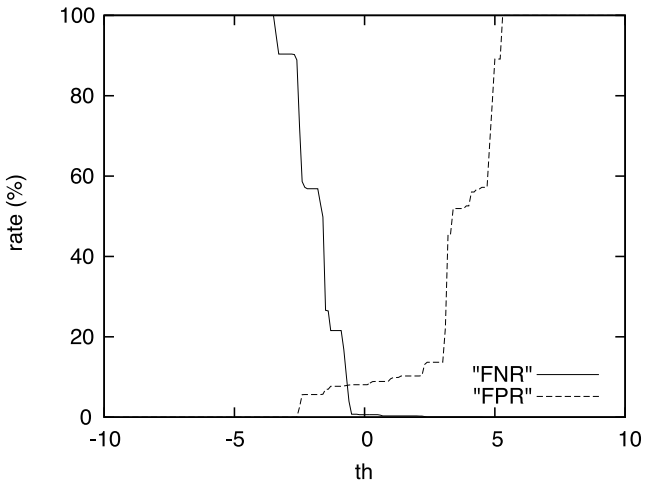

図 9 FPR と FNR の関係
Fig. 9 FPR and FNR.

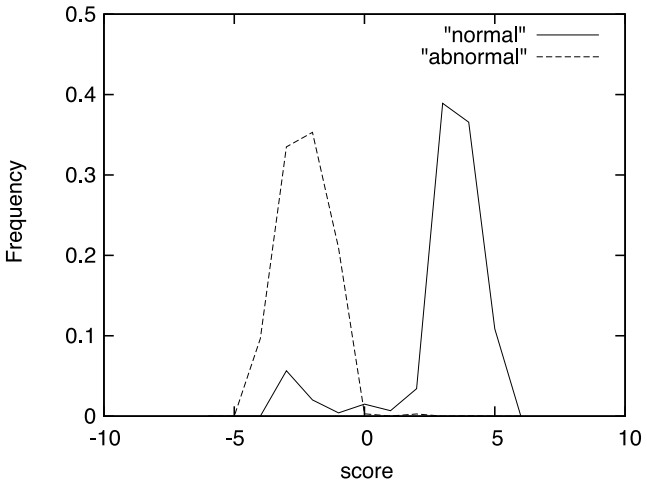

図 10 スコアのヒストグラム
Fig. 10 Histgram of scores.

図 10 において，th が小さい値，たとえば -10 のときにはすべてのスコアが th より大きくなるため感染のデータも正常と判定するので，図 9 において FNR = 100%，FPR = 0%となる．th を徐々に大きくしていくと th より大きくなる感染データのスコアが少なくなり，th より小さくなる正常データのスコアが多くなるので，FNR は小さくなり FPR は大きくなる．さらに th を大きくすると th より大きくなる感染データがなくなり FNR = 0%となる．さらに th を大きくするとすべての正常データのスコアが th より小さくなるので FPR = 100%となる．図 9 より，th を 1 つ決めるとそれに対応して FPR，FNR が定まるため，閾値 th を変えることにより FPR，FNR を制御できることが分かる．

また，図 9 より FNR，FPR が過敏に変化していることが分かる．ただし図 9 を見ると，FNR と FPR を同レベルで扱う場合には FNR と FPR が一致する th を設定することとなるが，FNR と FPR が一致する箇所付近（$th = 0$ 付近）は FNR，FPR が安定しており th を設定しやすいと考えられる．FNR もしくは FPR のどちらかを重視して th を設定する際には要求精度に応じて FNR，FPR の過敏な変化に注意して設定する必要がある．th の適切な設定については今後の課題としたい．

また，複数のスロットを使って感染検知を行う本提案手法においても式 (8) を

$$I(x) = sign[\Sigma_{t=1}^{T} s(x_t) + th] \tag{11}$$

のように th を導入できる．たとえば，th を負の値にすると，式 (11) の判定において負方向にスコアの伸びを大きくするため，より感染を検知しやすくするなどの制御が可能になると考えられる．

5.4 最終到達判定・先行判定について

単一スロットによる識別に比べ最終到達判定または先行判定を用いて識別することにより，高い識別性能を得ることができた．スロットごとのスコアを調べてみると，たとえば，正常通信データに対して大部分が正のスコアで正常と識別されているが，ところどころ負のスコアとなっている．このため，最終到達判定または先行判定を利用することにより高い識別性能が得られたと考えられる．

また，先行判定による識別について，図 6 より TH を大きくするにつれて感染検知率が上昇していることが分かる．さらに図 7 より TH を大きくするにつれて誤検知率が減少していることが分かる．ただし，TH が大きくなるにつれて TH に達するまでのスロット数がより多く必要になる．これより，感染検知率，誤検知率と必要となるス

ロット数にはトレードオフの関係があるので，早いタイミングで感染を警告したい場合には TH を低く設定し，感染検知の誤りを少なくしたい場合には TH を大きめに設定する必要がある．今回の実験において，図 6，7 より閾値 $TH = 3$ 以上のときに TPR = 100%，FNR = 0%となる．TH を大きくするにつれて必要となるスロット数が多くなるので処理時間が必要となる．その点も考慮して今回の実験については最適な閾値は TH=3 となる．

最終到達判定による識別（3.2 節）では，図 5 より T を大きくすると識別率が上がることが分かる．T スロットのスコア算出が終わってから識別を行うため，T の値を大きくすると正常トラヒックか感染トラヒックかを識別するまでに時間がかかる．その点を考慮して T の値を決めることとなる．今回の実験では T が 7 以上で識別率が 100%となっている．

また，今回の評価実験では 4.2 節で説明したデータに対して従来手法と提案手法を比較し，提案手法は従来手法よりも高い識別性能を得ることができた．本提案手法の有効な範囲を示すために，評価データを PCA などにより特徴空間での分布の様子を見ることでどの程度の複雑度であるのかを調べていく．また，今回の評価実験では学習用と評価用でトラヒックの取得時期が異なるデータでの評価を行った．ただし，学習データ取得の約 10 カ月後のデータで評価したのみなので，今後はさらに異なる時期に取得したデータを利用して識別精度がどのように変化するのかを評価する必要がある．さらに取得環境の異なる他のデータセットでの評価を行い，取得環境の違いについても評価する必要がある．

5.5 オンラインでの感染検知

本論文で報告したのは，オフラインで評価実験を行った実験結果である．ただし，実用を考慮すると，感染拡大を防ぐためにはオンラインでの感染検知が必要である．そこで本論文で述べた方法をオンラインでの感染検知にどのように生かすことができるのかについて考察した．

AdaBoost のスコアはベイズ則に一致するため正方向に絶対値が大きいほど正常時通信トラヒックらしいということができる．また，負方向に絶対値が大きいほど感染時通信トラヒックらしいということができる．そのことを利用して対象区間あたりのスコアの伸びに着目する．**図 11** に示すようにマルウェアに感染した場合，負方向のスコアの伸びが連続すると考えられる．連続的にスコアの伸びを算出し[*12]，連続する負方向のスコアの伸びを検知し，先行判定することで早期に感染検知ができる可能性があると考えられる．

[*12] あらかじめ AdaBoost により求めておいた強識別器にデータを入力しスコアを求めるのみなので，スコア算出のための時間はほとんどかからない．

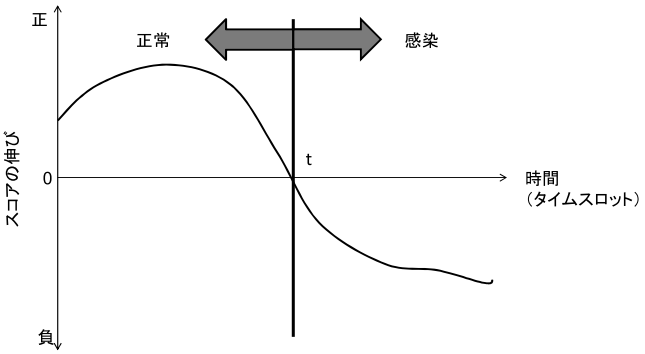

図 11 オンラインでの感染検知
Fig. 11 Concept of online detection.

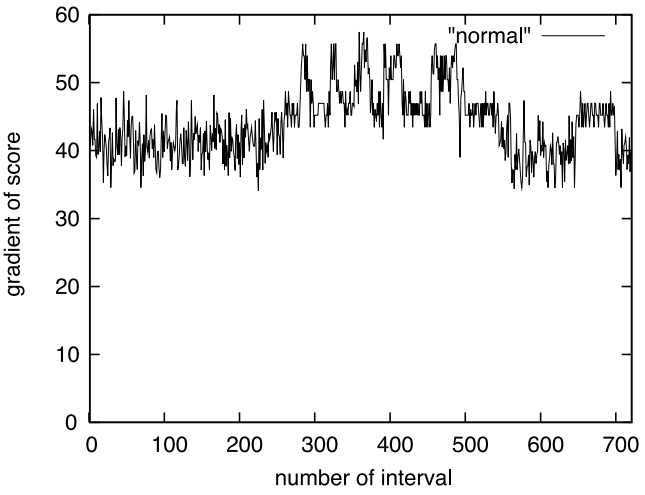

図 12 正常時通信のスコアの伸び
Fig. 12 Gradient of score (Normal communication).

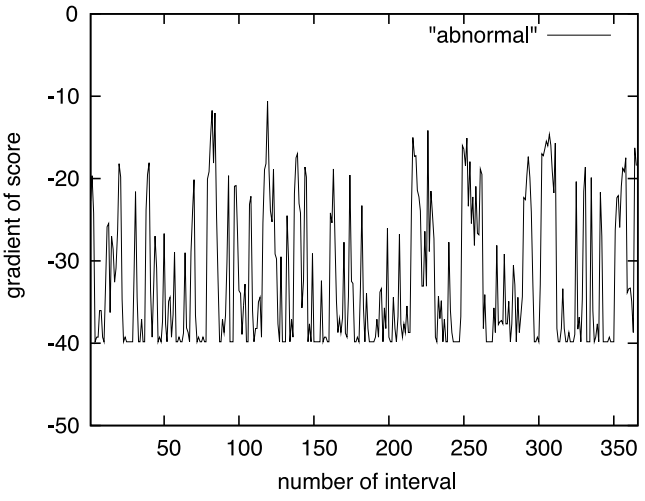

図 13 感染時通信のスコアの伸び
Fig. 13 Gradient of score (Abnormal communication).

4.3 節で示した実験での正常時通信トラヒックのスコアの伸び（式 (7) の $S(x)$）および感染時通信トラヒックのスコアの伸び（式 (7) の $S(x)$）を**図 12**，**図 13** に示す．横軸が対象区間の番号，縦軸がスコアの伸びを表す．これらの図より正常時通信の場合には正方向のスコアの伸びが連

続し，マルウェアに感染した後の通信の場合には負方向のスコアの伸びが連続することが分かる．これより連続する負方向のスコアの伸びを検知し，先行判定することで早期に感染を検知できる可能性があると考えられる．

5.6 提案手法の実用性

提案手法の実用性について考える．実用的かどうかについは，識別精度，パラメータ数，パラメータのチューニングのしやすさ，処理時間が重要になると考える．

識別精度については，表 2 より提案手法のポイントである AdaBoost のスコアはベイズ則に一致することを考慮し最終到達判定または先行判定により識別することが有効であると考える．

提案手法で必要となるパラメータは，特徴量，弱識別器の数，T，TH となる．特徴量については使用できる特徴量を用意すれば AdaBoost で選択できるので，実装するうえでチューニングする必要があるパラメータは弱識別器の数，T，TH となる．

弱識別器の数については，3.1 節でも述べたように弱識別器の数が大きいほど識別精度が良くなるという関係があるので，識別に使用する弱識別器の数を調整することにより識別時間と識別精度を調整することができる．つまり，要求される識別時間と識別精度より弱識別器の数を選ぶことができるのでチューニングしやすいと考えられる．T，TH についても，5.4 節より識別時間と識別精度のトレードオフの関係があるので要求される識別時間と識別精度より T，TH の値を設定することができるのでチューニングしやすいと考えられる．

処理時間については，学習に使用するサンプル数を m，弱識別器の数を B，使用する特徴量の種類を F とすると，学習時の処理時間は $O(FmB)$ となる．また，識別時の処理時間は $O(BT)$ となる．サンプル数が増えた場合においても 1 次のオーダでの処理時間の増加となり，2 次以上のオーダのような急激な増加とはならない．また，あらかじめ学習時に強識別器を作成しておくため，識別時ではその強識別器を用いてスコアを算出するのみなので処理時間はほとんどかからない．

以上より，提案手法は実用的と考える．

6. むすび

本論文では，トラヒックデータの連続入力を考慮した，AdaBoost に基づくマルウェア感染検知手法を提案した．そして，研究用データセット CCCDATASet を用いた実験で，先行判定と最終到達判定を組み合わせることによって提案手法がうまく機能することを確認した．さらに先行判定には早いタイミングで感染を検知し，警告を知らせることができる可能性があることを確認した．

提案手法の有効性を確認するために，さらに異なる時期に取得したデータを利用して識別精度がどのように変化するのかを評価する．さらに取得環境の異なる他のデータセットでの評価を行い，特徴空間の分布の様子を調べながら有効性を確認していきたい．

今後は，トラヒックデータは様々な挙動の組合せでできていることをふまえ，トラヒックデータの状態遷移も考慮した感染検知手法を検討し，今回検討した先行判定・最終到達判定と組み合わせた検知手法を検討する予定である．

参考文献

[1] PART.3 犯行に協力させる攻撃サイト改ざんの手法に変化仮想サーバーが復旧に役立つ，日経 NETWORK，pp.35–39 (2011).

[2] 山城重成：多様化する脅威に対応した今どきのセキュリティ機能，日経 NETWORK，pp.76–79 (2010).

[3] Masud, M.M., Al-khateeb, T., Khan, L., Thuraisingham, B. and Hamlen, K.W.: Flow-based Identification of Botnet Traffic by Mining Multiple log Files, *1st International Conference on Distributed Framework and Applications*, pp.200–206 (2008).

[4] Lakhina, A., Crovella, M. and Diot, C.: Mining Anomalies Using Traffic Feature Distributions, *SIGCOMM'05*, pp.217–228 (2005).

[5] Yen, T.-F. and Reiter, M.K.: Traffic Aggregation for Malware Detection, *DIMVA2008*, LNCS5137, pp.207–227 (2008).

[6] Parekh, J.J., Wang, K. and Stolfo, S.J.: Privacy-preserving payload-based correlation for accurate malicious traffic detection, *the ACM SIGCOMM Workshop on Large Scale Attack Defense*, pp.99–106 (2006).

[7] Karamcheti, V., Geiger, D., Kedem, Z. and Muthukrishnan, S.M.: Detecting malicious network traffic using inverse distributions of packet contents, *the ACM SIGCOMM Workshop on Mining Network Data*, pp.165–170 (2005).

[8] Wang, K., Cretu, G. and Stolfo, S.J.: Anomalous Payload-Based Worm Detection and Signature Generation, Lecture Notes in Computer Science, Vol.3858/2006, pp.227–246 (2006).

[9] Kondo, S. and Sato, N.: Botnet Traffic Detection Techniques by C&C Session Classification Using SVM, *IWSEC2007* (2007).

[10] Livadas, C., Walsh, B., Lapsley, D. and Strayer, T.: Using Machine Learning Techniques to identify botnet traffic, *Proc. 2nd IEEE LCN Workshop on Network Security* (2006).

[11] Saad, S., Traore, I., Ghorbani, A., Sayed, B., Zhao, D., Lu, W., Felix, J. and Hakimian, P.: Detecting P2P botnets through network behavior analysis and machine learning, *2011 9th Annual International Conference on Privacy, Security and Trust* (*PST*), pp.174–180 (2011).

[12] Li, W. and Li, Q.-X.: Using Naive Bayes with AdaBoost to Enhance Network Anomaly Intrusion Detection, *3rd International Conference on Intelligent Networks and Intelligent Systems* (*ICINIS*), pp.486–489 (2010).

[13] Hu, W., Hu, W. and Maybank, S.: AdaBoost-Based Algorithm for Network Intrusion Detection, *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol.38, No.2, pp.577–583 (2008)

[14] Zhang, J., Liu, Y. and Liu, X.: Anomalous Detection Based on Adaboost-HMM, *The 6th World Congress*

on Intelligent Control and Automation, pp.4360–4363 (2006).
[15] Farid, D.Md., Rahman, M.Z. and Rahman, C.M.: Adaptive Intrusion Detection based on Boosting and Naive Bayesian Classifier, *International Journal of Computer Applications*, Vol.24, No.3, pp.12–19 (2011).
[16] Chen, Y.-S. and Chen, Y.-M.: Combining Incremental Hidden Markov Model and AdaBoost Algorithm for Anomaly Intrusion Detection, *CSI-KDD'09*, *Proc. ACM SIGKDD Workshop on CyberSecurity and Intelligence Informatics*, pp.3–9 (2009).
[17] Kishimoto, K., Yamaki, H. and Takakura, H.: Improving Performance of Anomaly-Based IDS by Combining Multiple Classifiers, *Applications and the Internet* (*SAINT*), *2011 IEEE/IPSJ 11th International Symposium*, pp.366–371 (2011).
[18] 川元研治，市田達也，市野将嗣，畑田充弘，小松尚久 :マルウェア感染検知のための経年変化を考慮した特徴量評価に関する一考察，マルウェア対策研究人材育成ワークショップ 2011（MWS2011）(2011).
[19] Freund, Y. amd Schapire, R.E.: A decision theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Science*, Vol.55, No.1, pp.119–139 (1997).
[20] 三田雄志 :AdaBoost の基本原理と顔検出への応用 CVIM 研究会チュートリアルシリーズ，CVIM, Vol.159, pp.265–272 (2007).
[21] 畑田充弘，中津留勇，秋山満昭 :マルウェア対策のための研究用データセット—MWS 2011 Datasets，マルウェア対策研究人材育成ワークショップ 2011（MWS2011）(2011).
[22] available from ⟨http://svmlight.joachims.org/⟩.
[23] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I.H.: The WEKA Data Mining Software: An Update, *SIGKDD Explorations*, Vol.11, No.1 (2009).

市野 将嗣 （正会員）

2003 年早稲田大学理工学部電子・情報通信学科卒業．2008 年同大学大学院理工学研究科博士課程修了．2007 年日本学術振興会特別研究員．2009 年早稲田大学大学院基幹理工学研究科研究助手．2010 年同大学メディアネットワークセンター助手．2011 年電気通信大学大学院情報理工学研究科助教．バイオメトリクス等パターン認識，ネットワークの品質と安全性を考慮したトラヒック分類に関する研究に従事．博士（工学）．電子情報通信学会会員．

市田 達也

1987 年生．2012 年早稲田大学理工学術院基幹理工学研究科情報理工学専攻修士課程修了．同年 NTT コミュニケーションズ株式会社入社．在学中はマルウェア感染検知に関する研究に従事．

畑田 充弘 （正会員）

2003 年早稲田大学大学院理工学研究科修士課程修了．同年 NTT コミュニケーションズ（株）入社．以来，マルウェア対策をはじめとするネットワークセキュリティの研究開発に従事．

小松 尚久 （正会員）

1979 年早稲田大学理工学部電子通信学科卒業．1981 年同大学大学院理工学研究科修士課程了．同年 NTT に入社．1987 年早稲田大学理工学部助手．1989 年同専任講師．1996 年同教授．工学博士．現在，セキュリティと品質を考慮したヒューマン・ネットワークインタフェースに関する研究に従事．電子情報通信学会，画像電子学会，IEEE 各会員．