トラヒックの時系列データを考慮した マルウェア感染検知手法に関する一検討

市野 将嗣 †　　市田 達也 ††　　畑田 充弘 †††　　小松 尚久 ††

† 電気通信大学大学院情報理工学研究科総合情報学専攻
182-8585 東京都調布市調布ヶ丘 1-5-1
ichino@inf.uec.ac.jp

†† 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
169-8555 東京都新宿区大久保 3-4-1
{ichida,komatsu}@kom.comm.waseda.ac.jp

†††NTT コミュニケーションズ株式会社
〒 108-8118 東京都港区芝浦 3-4-1 グランパークタワー 17F
m.hatada@ntt.com

あらまし 本研究では，トラヒックの時系列データを考慮した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データを正常時の通信トラヒックデータと比較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 の特徴を踏まえた時系列データの感染検知手法について検討した．本稿では，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について報告する．

A study on malware detection method using time series traffic data

Masatsugu Ichino†　　Tatsuya Ichida††　　Mitsuhiro Hatada†††
Naohisa Komatsu††

†Graduate School of Informatics and Engineering, University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182-8585 Japan
ichino@inf.uec.ac.jp

‡Faculty of Science and Engineering, Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555, Japan
{ichida,komatsu}@kom.comm.waseda.ac.jp

†††NTT Communications Corporation
Gran Park Tower 17F, 3-4-1 Shibaura, Minato-ku, Tokyo, 108-8118 Japan
m.hatada@ntt.com

Abstract We propose a method of malware detection method using time series traffic data. Damage by malware attack has been viewed with suspicion recently. We studied the malware detection method by comparing malware traffic with normal traffic. 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 method using time series traffic data. In this paper, we evaluated the effectiveness of proposed method by using CCCDATASet.

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ーに代表される 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．

これまでの対策研究としては，文献 [1] で整理されているように，PC がマルウェアに感染しているかどうかを検知するためのマルウェアの感染検知，感染後の挙動の観測やコードの解析を行うマルウェアの検体解析，ハニーポットを用いたボットネット等の活動状況の観測を行う広域観測といった視点で研究が行われている．

近年のマルウェアによる感染は気付きにくいため，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感染検知は重要な対策である．しかしながら，従来の感染検知では，既知のマルウェアの検知が中心となっており，一方で過去のマルウェア挙動から未知のものを予測で検知する手法も存在するが，誤検知がある．そこで本研究では，感染後の通信トラヒックから感染していることを早期に検知することを考える．正常時通信トラヒック，マルウェア感染後のトラヒックの特徴を捉え，さらにトラヒックは時系列データであり様々な挙動の組み合わせであることを踏まえ，正常時通信トラヒック，感染時通信トラヒックを段階的に識別することで未知のマルウェアによる感染も検知することを目指す．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する．

そこで本稿では，トラヒックデータの連続入力を考慮した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** 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．**3** 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．**4** では，実験結果に対する考察を述べる．**5** はまとめと今後の課題である．

2 感染検知システム

本章では，マルウェアの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，連続入力を考慮した複数スロットによる識別方法) について説明する．

2.1 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．

フローの統計情報を用いた識別手法では，複数のパケットをフローという単位で一纏めにして，フローにおけるパケットサイズの平均値，標準偏差等と加工して利用している．つまり，連続的な入力が仮定できるにも関わらず，トラヒックの時間的な変化を一纏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．また，プライバシー保護の観点からもヘッダ情報から得られる統計情報で感染検知できることが望ましい．

2.2 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 [2][3]．ナイーブベイズは，あらかじめ事前確率がわ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ナイーブベイズを適用することは適切ではないと考えられる．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[1]．SVM は調整可能なパラメータがあるため識別制御が可能である．SVM はマージン最大化の基準により識別関数[2] $w^T\Psi(x)+b$ のうち w と b が決まる (w と b に自由度はない)．このスコアが 0 のところを閾値として識別する場合が多いが，閾値を動かすことにより閾値制御を行うことも可能である．ただしこの閾値を動かす意味をマージン最大化基準によるアルゴリズムから説明がつきづらいので，閾値制御の根拠が薄いと考えられる．また，SVM はカーネルパラメータやソフトマージンパラメータのようなチューニングを必要とするパラメータを複数もっているため，多くの試行回数のパラメータ探索が必要となる．また，ナイーブベイズ，SVM は識別に使用する特徴量をあらかじめ決

[1] 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，FP(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等の PC やサーバでは，FN(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．

[2] Ψ を関数空間への非線形写像，$\vec{w}$ を 1 次元空間への変換を表す射影ベクトルとする

めておく必要がある．ただし，ヘッダ情報から得られる統計情報は多くあり (文献 [4] では 36 個)，また，トラヒックは様々な挙動の組み合わせであり，挙動によって有効な特徴量は異なると考えられるため選択するのは難しい．決定木は，識別ルールが視覚的にわかり便利であり，エントロピーの減少が最大になる属性を選択することにより特徴量の選択も行われるが，一撃で識別面を作成するため閾値制御の根拠が薄い．

以上の問題点を踏まえ，本研究ではマルウェア感染検知を行うために AdaBoost[5][6] を適用する．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み合わ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 [4] 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 FP,FN 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．

図 1 に AdaBoost を適用する際の概要を示す．図 1 のように各スロットにおける識別において，サンプル[5]の重み分布 D_b に基づき各特徴量の弱識別器 h_b と各特徴量 (弱識別器) の重み (信頼度 α_b) を求め，学習サンプルに対する誤り率

$$\epsilon_b = \Sigma_{i:y_i \neq h_b(x)} D_b(i) \tag{1}$$

が最小となる特徴量 (弱識別器) を選択する．弱識別器を以下のように設計する．

$$h_b(x) = \begin{cases} +1 & if \quad p \cdot s_t(x) > p \cdot \theta \\ -1 & otherwise \end{cases} \tag{2}$$

ただし，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．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．

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．

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

$$H(x) = sign[\Sigma_{b=1}^{B} \alpha_b h_b(x) + th] \tag{3}$$

式 (3) の th を調整することでセキュリティレベルに応じた閾値制御を行うことができる．$H(x)$ はベイズ則と一致するため，AdaBoost のスコアを

$$s_t(x) = \Sigma_{b=1}^{B} \alpha_b h_b(x) \tag{4}$$

とする．

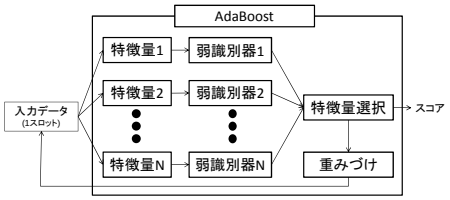

図 1: AdaBoost の概要

2.3 複数スロットによる識別

感染検知を実用化するためにはさらなる精度向上とともに早期に感染を検知することが重要となる．複数スロットを使うことでこれらを実現する方法について検討した．

2.2 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

[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]本稿では，1 スロットを 1 サンプルとする．

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，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具合) に着目する．複数スロットによる識別の概要を図 2 に示す．フローに基づく方法ではフローが収集終わるまで検知できないのに対して，スロットごとに求まるスコアの伸びの大きさを見ることで感染していることを早く検知して警告を知らせることができる．

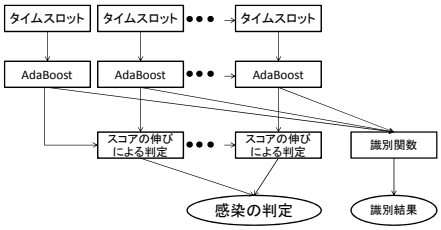

図 2: 複数スロットによる識別の概要

本稿での連続するスロット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．本稿では，AdaBoost によるスコアはベイズ則に一致することを踏まえ識別関数を

$$S(x) = sign[\Sigma_{t=1}^{T} s_t(x)] \qquad (5)$$

のように表し，識別対象区間 (区間幅を T とする) における各スロットのスコアの和をスコアとして扱う (図 3 の ②)．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，早期の感染検知については，$t < T$ の間で，スコアの和が $-TH$ 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)．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．

3 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．

3.1 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 (特徴

[6]1 スロットが特徴空間における 1 点に相当する

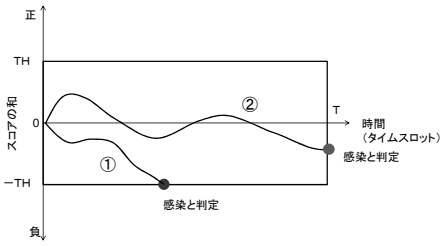

図 3: 複数スロットによる識別のイメージ

空間における識別面) 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．スコアの伸び判定では，時刻が T 以下で，各スロットのスコアの和が $-TH$ を下回った時に感染と判定する．複数スロットでの識別では，時刻が 0 から T までのスロットのスコアの和が正であれば正常，負であればマルウェアに感染していると判定する．

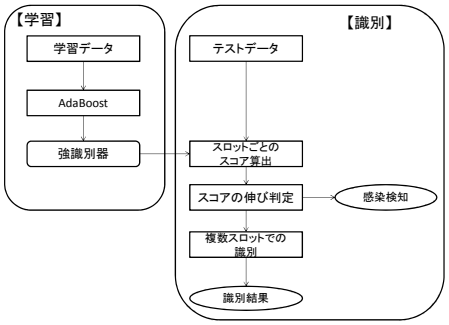

図 4: 実験系の概要

3.2 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[7] を用いた．

CCC2010,2011 のデータ取得期間に合わせて，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，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．学習データとテストデータの組み合わせについて

1. 学習データとテストデータの取得が同時期の組み合わせ (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)　テストデータ:36566 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み合わせ (学習データ:2010 年，テストデータ:2011 年)　テストデータ:44843 スロット

の 2 パターンの実験を行った．

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=2 分 (図 3 参照) とした．本実験では，文献 [4] で示されている 36 個の特徴量を用いた．

3.3　実験結果

AdaBoost を用いることの有効性を確認するために SVM との比較を行った．AdaBoost については，予備実験により $B = 10$ とした．SVM については，特徴量を 36 個使用し，カーネル関数はガウス型動径基底関数

$$k(\vec{x}, \vec{y}) = \exp\left(\frac{-||\vec{x} - \vec{y}||^2}{2\sigma^2}\right) \qquad (6)$$

を用い，予備実験により適切と思われる σ を設定し，SVM^{light}[8] を使用して識別を行った．学習データとテストデータの取得が同時期，異時期の組み合わせの識別結果[7]を表 1 に示す．

また，複数スロット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．表 1 に，単一のスロットごとに識別した結果 (AdaBoost(単一スロット)) と **2.3** で示した複数スロットを考慮した識別の結果 (AdaBoost(複数スロット)) もあわせて示す．

表 1 より AdaBoost は SVM より TPR,TNR が安定しており，高い識別率が得られている．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．

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 HTTPS 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を，使用する順番を変えて利用している．文献 [4] の評価実験において，これらの特徴量は 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．

[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．

また，表 1 より複数スロットを使用した識別は単一の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わかる．

スコアの伸び判定について伸び判定する閾値 TH(図 3 の TH) に対する感染検知率[8]および使用した平均スロット数の関係を図 5 に示す．また，伸び判定する閾値 TH に対する誤検知率[9](FN) および使用した平均スロット数の関係を図 6 に示す．閾値 TH が小さいときに高い感染検知性能が得られているため，早いタイミングで感染を警告することができる可能性がある．

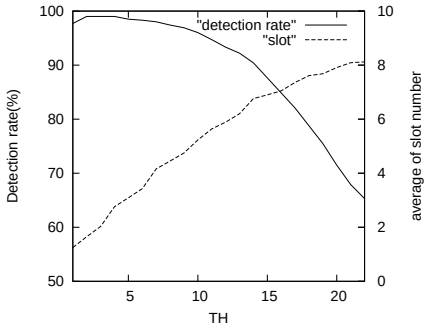

図 5: 閾値 TH に対する感染検知率と平均スロット数の関係

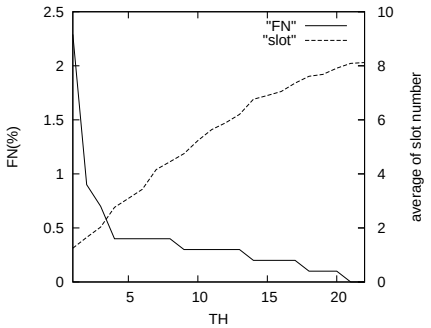

図 6: 閾値 TH に対する誤検知率 (FN) と平均スロット数の関係

4　考察

4.1　複数スロットによる識別

単一スロットによる識別に比べ複数スロット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，例えば，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，複数スロット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，スコアの伸び判定について図 6 より閾値 TH を大きくするとスコアが $-TH$ に到達するまでのスロット数が大きくなり，それに伴っ

[8]感染データを感染していると検知する割合．

[9]感染データのうちスコアの和が $+TH$ より大きくなった割合．

表 1: 識別結果

	同時期			異時期		
識別手法	識別率 (%)	TPR(%)	TNR(%)	識別率 (%)	TPR(%)	TNR(%)
SVM	91.0	67.7	99.6	92.2	41.8	99.9
AdaBoost(単一スロット)	98.6	95.2	99.9	92.9	91.9	99.4
AdaBoost(複数スロット)	99.8	99.3	100.0	100.0	100.0	100.0

て感染の誤検知率が低下していることがわかる．一方図 5 より，閾値 TH を大きくすると，スコアの伸び判定では検知できなくなり感染検知率が低下している．これは，$-TH$ に到達するまでのスロット数がより必要になるため感染検知まで至っていないためである．ただし，感染検知したものについては図 6 の結果からわかるようにほとんど正しく感染データを検知している．T を大きくすると使用できるスロット数が増えるため感染検知率が上がると考えられる．

4.2 オンラインでの感染検知

本稿で報告したのは，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．そこで本稿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単位区間あたりのスコアの伸びに着目する．図 7 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[10]，連続する負方向のスコアの伸びを検知することで早期に感染検知ができると考えられる．

5 むすび

本稿では，トラヒックデータの連続入力を考慮し，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 を用いた実験で有効性を示した．スコアの伸びによる判定においても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．

今後は，他の識別アルゴリズムとの比較を行い AdaBoost を用いることの有効性を確認していく．またスコアの伸びの表現方法についても検討していく．さらに他の研究用データセットを使用した評価実験も行い有効性を確認したい．

[10]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．

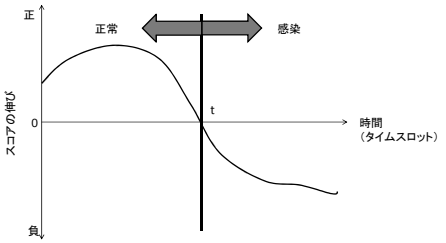

図 7: オンラインでの感染検知

参考文献

[1] 藤原将志，寺田真敏，安部哲哉，菊池浩明，“マルウェアの感染方式に基づく分類に関する検討,” 情報処理学会 CSEC 研究報告，No.21，p177-182，March 2008.

[2] S.Kondo and N.Sato, “Botnet Traffic Detection Techniques by C&C Session Classification Using SVM,” IWSEC2007, October 2007.

[3] Livadas C., Walsh B., Lapsley D., Strayer T, “Using Machine Learning Techniques to identify botnet traffic,” In Proceedings of 2nd IEEE LCN Workshop on Network Security, November 2006.

[4] 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)，October 2011(発表予定)

[5] Y. Freund, R. E. Schapire, “ A decision theoretic generalization of on-line learning and an application to boosting,” Journal of Computer and System Science, Vol. 55(1), pp. 119-139, 1997.

[6] 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，May 2007.

[7] 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)，October 2011.

[8] http://svmlight.joachims.org/