情報処理学会論文誌 Vol.56 No.9 1770–1781 (Sep. 2015)

挙動に基づくポートスキャン検知の自動化に向けた学習アルゴリズムの提案とその性能評価

王 サン[1,2,†1] フォン ヤオカイ[1,2,a)] 川本 淳平[1,2] 堀 良彰[2,3] 櫻井 幸一[1,2]

受付日 2014年12月5日, 採録日 2015年6月5日

概要：近年，挙動に基づく検知手法はインターネット上の攻撃を検知する手法として注目を浴びてきた．この手法は学習データから抽出した通常モードを利用して異常検知を行う．そのため，他の閾値を用いた手法と異なり事前に通常と異常を区別する閾値を決める必要がない．通常モードを抽出には，事前に与えられた度数分布図に対して学習アルゴリズムを適用する．しかしながら，既存研究では度数分布図に対する学習アルゴリズムにおいて，パラメータチューニングが必要であった．本研究では挙動に基づく検知手法において，パラメータなしの学習アルゴリズムを提案する．また，実験検証により，本提案の学習アルゴリズムはインターネット上の攻撃の検知に有効であることを示す．

キーワード：ポートスキャン，挙動に基づく検知法，サイバー攻撃

A Learning Algorithm for Behavior-based PortScan Automatic Detection and Its Evaluation

Can Wang[1,2,†1] Yaokai Feng[1,2,a)] Junpei Kawamoto[1,2] Yoshiaki Hori[2,3] Kouichi Sakurai[1,2]

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

Abstract: In recent years, behavior-based methods have attracted many researchers in the field of cyber-attack detection. As such methods exploit the normal behavior modes extracted from learning data to detect anomalies, it is no longer necessary to set thresholds in advance that are used to distinguish normal and abnormal traffic. For extracting the normal behavior modes, a learning algorithm is often used after the frequency diagrams have been built. However, even the frequency diagrams have been drawn, there are still some parameters need to be determined in advance in the existing learning algorithms for extracting normal modes from the frequency diagrams and such parameters are often not easy to decide in advance. To solve this problem, we propose a novel learning algorithm, in which no parameters need to be tuned. According to our discussion and experimental results, our proposed learning algorithm is efficient for detecting cyber-attacks.

Keywords: PortScan, behavior-based detection, cyber attacks

1. はじめに

近年，インターネットの利用率がますます高くなっている．総務省によると，平成 25 年インターネットの利用率は 82.8%であり，近年の最大値となっている．その中で 81.4%の利用者は個人情報の保護に不安を感じている [1]．そのため，安全・安心なインターネット環境の構築は重要

[1] 九州大学大学院システム情報科学研究院
Graduate School of Information Science and Electrical Engineering, Kyushu University, 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] 佐賀大学全学教育機構
Organization for General Education, Saga University, Saga 840–8502, Japan
[†1] 現在，東日本電信電話株式会社
Presently with NTT East Corporation
[a)] fengyk@ait.kyushu-u.ac.jp

本論文は，国際会議である AsiaJCIS2014 で発表した論文の増補版として，大幅に加筆した．

な課題である．特に，秘密情報が個人や企業に気づかれることなく不正者に窃取されると，大きな損失が発生しうる．このため，多くの研究者がインターネット上の攻撃に関する対策研究を行ってきた．しかしながら，発生している攻撃の検知および対抗や潜在的な脅威予測はまだ大きな課題である．インターネット上の攻撃を検知するために，様々な手法が提案されてきた．IDS（侵入検知システム）や IPS（侵入阻止システム）は，実社会でもよく利用されている．IDS は IP パケットをフィルタリングすることで不審なアクセスをリアルタイムに検知するシステムである [2]．IDS には，シグネチャ型 IDS や異常検出型 IDS などの種類がある．シグネチャ型 IDS では事前に設定されるルールで不正侵入を検知する．適切なルールを設定できれば高い検知率を達成できるというメリットがあるが，未知の攻撃への対応が難しいという限界がある [3]．異常検出型 IDS は通常モードを抽出し，それを用いて異常検知を行う．通常モードを適切に抽出できれば，未知の攻撃を検知できる．異常検出型 IDS に用いられている検知手法として，Denning は挙動に基づく検知手法を提案した [4]．

挙動に基づく検知手法にはいくつかの利点がある．たとえば，新種や変種攻撃への対応が可能であることや，抽出した通常モードは監視する実際のネットワークのトラフィック特徴を反映し，実際の通信状況によって通常モードは自動的に更新するため，環境ごとに適切な検知が可能であることがあげられる．さらに，同じネットワークに対しても，複数の通常モードを抽出すれば，異なる状況にも対応できる．この長所から，挙動に基づく検知手法はよく注目されてきた．挙動に基づく検知手法では，通常モードを抽出する学習アルゴリズムは核心な役割をになっている．しかしながら，既存の学習アルゴリズムでは，いくつかのパラメータを事前に設定する必要があった．本研究では，挙動に基づく検知手法のために，初めてパラメータチューニングが要らない学習アルゴリズムを提案する．実験の結果および討議により，本研究で提案する学習アルゴリズムを利用して過去の観測データから抽出された通常モードは信頼性と応用性があることが分かる．

本論文の構成は以下に示す．2 章はポートスキャンに関する予備知識を紹介する．3 章は最新の既存学習アルゴリズムを紹介する．4 章は本研究の中核な部分として，パラメータチューニングを省いた学習アルゴリズムを提案する．5 章は実験結果を示す．最後に 6 章に結論と今後の課題について述べる．

2. ポートスキャンと挙動に基づく検知

2.1 ポートスキャン

ある計算機への侵入を目論む不正者は，初めに標的ホストの脆弱性情報を収集することが多い [6]．そのために，ポートスキャンという技術がよく用いられる．ポートスキャンとは，ターゲットのポートをスキャンし，ターゲットで動作しているアプリケーションソフトウェアや OS の種類を調べ，侵入口となりうる脆弱なポートがあるか調べる行為である．ポートスキャンによって脆弱性を発見すると，実際の攻撃に移ることが多い．そのため，ポートスキャン検知は攻撃の早期発見のために課題となってきた．

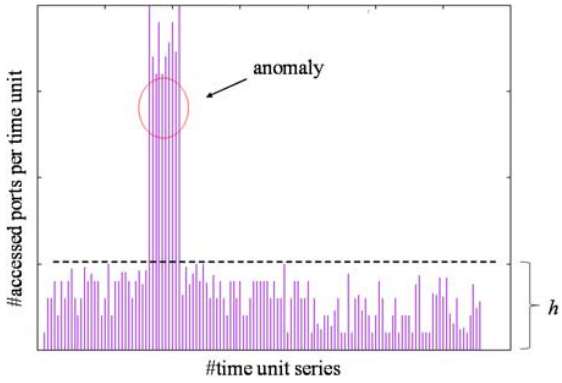


図 1 挙動に基づく手法の一例
Fig. 1 Example of behavior-based anomaly detection.

ポートスキャンは一般的に垂直ポートスキャンと水平ポートスキャン 2 種に大別される．垂直ポートスキャンは 1 台の標的ホストに対していくつか（もしくはすべて）のポートをスキャンする行為であり，水平ポートスキャンはあるポートに対して，複数のホストをスキャンする行為である．さらに，攻撃始点ホストの数により，単一始点ホストのポートスキャンおよび分散型ポートスキャンの 2 種類に分けられる．本研究では，垂直ポートスキャンの検知を目指す．

2.2 挙動に基づくポートスキャン検知

図 1 は挙動に基づくポートスキャン検知手法の概念を示したものである．横軸は時間単位の時系列で，縦軸の定義は応用先により異なる．この例では，縦軸は各時間単位にアクセスされた終点ポート数を示す．図 1 から分かるように，通常な状況ではアクセスされた終点ポートの数がある値 h を超えない．一方，丸で囲まれるトラフィックでは時間単位ごとにアクセスされた終点ポートの数が特に大きいため，それを異常と見なす．この場合 h は通常モードと呼ばれる．もし過去のデータから h の範囲を抽出できれば，異常を簡単に検知できる．したがって，過去のデータから通常モードの抽出は，挙動に基づくポートスキャン検知の核心になる．十分な時間にわたるノイズがないデータがあれば通常モードは簡単に抽出できるが，実際の場合ではそのようなデータの入手は難しい．そのため，ノイズがある実際の状況でも通常モードを抽出することができる学習アルゴリズムが必要である．Feng ら [5] はこのような要求を満たす学習アルゴリズムを提案した．本研究ではその学習アルゴリズムを FHST 学習アルゴリズムと呼び，3 章で詳

しく紹介する．

挙動に基づくポートスキャン検知手法の一般的な流れを**表 1** で示す．ステップ 1 では学習用トラフィックデータを収集し，ステップ 2 で各時間単位内にアクセスされたポート数を集計する．その後，ステップ 3 として度数分布を作成する．ステップ 4 は，学習アルゴリズムを利用して度数分布から通常モードを抽出する．本研究では，抽出した通常モードは通常時に与えられた時間単位内にアクセスされたポートの最大数を用いる．すなわち，時間単位内にアクセスされたポート数がそれより少ない場合は異常でないとされ，それより大きい場合はアラートを出す．最後のステップ 5 では現在のトラフィックデータと抽出した通常モードを比較することにより異常検知を行う．

以上の流れから分かるように，ステップ 4 は核心である．したがって，本研究はステップ 4 の通常モード抽出を注目する．

通常モードを抽出するための度数分布の作成について簡単に説明する．まず，与えられた時間単位内にアクセスされたポート数の範囲は量子化され，bin の形になり，横軸で表示する．1 つの bin は 1 つの数値範囲に対応する．次は各 bin に対して発生した時間単位数を集計し縦軸にする．**図 2** は度数分布図の一例である．各 bin の横軸の値は対応している数値範囲の始点である．たとえば，範囲 40〜49 の bin の高さが 40 である場合，学習トラフィックデータにおいて，40〜49 種類のポート数にアクセスがあった時間単位数が 40 であったことを表している．

表 1 挙動に基づくポートスキャン検知
Table 1 Behavior-based portscan detection.

1. 学習データを収集
2. 各時間単位内にアクセスされたポート数を集計
3. 度数分布図を作成
4. 学習アルゴリズムを用いて度数分布図から通常モードを抽出
5. 異常検知

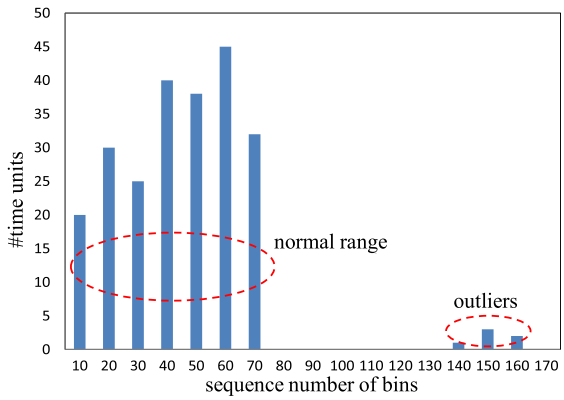


図 2 時間単位にアクセスされたポート数の度数分布
Fig. 2 Frequency distribution of numbers of accessed ports.

ポートスキャンの検知に関連する研究は多くある [11]．トラフィックデータをクラスタリングする手法 [12] や，通信パケットを見ながら検知モデルを作る手法 [13] などが提案されている．一般的な検知手法の多くでは閾値を決定する必要がある [14], [15], [16]．しかし，適当な閾値を決定するのは容易ではない．さらに，同一のネットワークであっても，時間の経過にともない，適切な閾値が変化する場合はよくある．Feng らは学習アルゴリズムを提案することより，過去のトラフィックデータから自動的に閾値を抽出することができた（詳細は 3 章を参照）．しかも，時間が経つと，通常モードを自動的に更新することも可能である．しかしながら，FHST 学習アルゴリズムは，2 つのパラメータが必要であり，その決め方は難しいという問題点がある．

3. FHST 学習アルゴリズム

Feng らは挙動に基づく異常検知のためのアルゴリズム（FHST 学習アルゴリズム）を提案している [5]．その学習アルゴリズムが利用する度数分布図の一例を**図 3** に示す．ここで，横軸は図 2 と異なっている．学習アルゴリズムの概要を**表 2**（初期化部分）と**図 4**（主体部分）に示す．

簡単にいえば，FHST 学習アルゴリズムは，度数分布を利用して通常モードを抽出する際に，一番右側の bin から

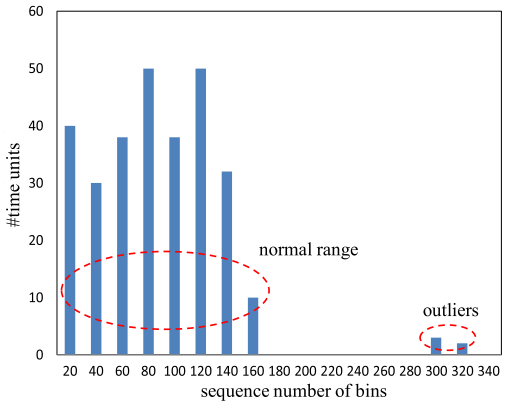


図 3 Feng ら提案で用いた度数分布図
Fig. 3 Frequency distribution used in Feng et al. [5].

表 2 FHST 学習アルゴリズム初期化 [5]
Table 2 Initiation of Feng algorithm [5].

Input: Frequency Distribution of the number of accessed different ports in one time unit **Output**: Normal behavior mode	
	Descriptions
Initializing	**Input**: Frequency Distribution diagram **Output**: normal behavior mode ***α%***: a threshold ***β%***: a threshold ***d***: distance to the next bin ***Ω***: group of checked bins ***Area(Ω)***: the number of time units in Ω

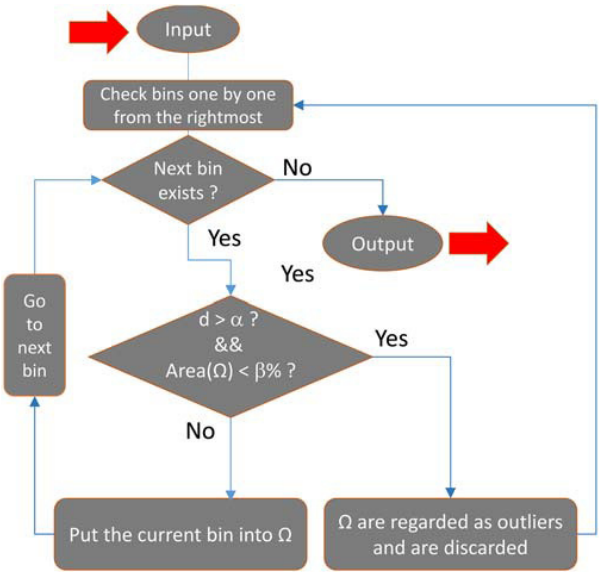


図 4 FHST 学習アルゴリズム [5]
Fig. 4 FHST algorithm [5].

bin を 1 個ずつ調べる．もし右側には十分に遠い（パラメータ α を利用）かつ面積（bin の高さの和）が十分に小さい（パラメータ β を利用）bin の集合が存在すれば，それらを学習データ中の異常（outlier）として削除した後，残った部分を通常と見なす．

このアルゴリズムの問題点として，2 つのパラメータ（α と β）が必要であり，その決め方はデータ依存であり一定不変のものではないので，専門家にとっても決定が難しいことである．本研究では，この問題を解決するためにパラメータなしの新しい学習アルゴリズムを提案する．

4. パラメータなしの学習アルゴリズムの提案

本提案の学習アルゴリズムはすでに我々の先行研究 [25] で紹介された．その発展として，本論文では提案学習アルゴリズムの学習性能を評価し，ポートスキャン検知に応用する際の検知性能を具体的に検証する．

4.1 アイディア

提案の学習アルゴリズムの目的は度数分布からパラメータを用いず通常モードを自動的に抽出することである．2 章で述べたように本研究では垂直ポートスキャンの検知を取り上げる．垂直ポートスキャンの場合は，多くのポートがアクセスされると考えられる．すなわち，垂直ポートスキャンのトラフィックは度数分布の右側に，通常モードのトラフィックは左側に集まることになる．そのため，度数分布図の両側から内側へすべての bin をチェックしながら通常 bin-group と異常 bin-group を区別できれば学習結果が得られる．ここで bin-group は度数分布の中で zero-bin（高さが zero である bin）により分けられた bin の集合を指す．本提案の学習アルゴリズムは 2 つのポインタを度数分布の両側に設置し，各々を内側に向かって移動させ，出会った点を学習結果とする．

表 3 本研究の学習アルゴリズム初期化
Table 3 Initiation of our proposal in this study.

Input: Frequency Distribution of the number of accessed different ports in one time unit **Output**: Normal behavior mode	
	Descriptions
Initializing	**Left_Pointer**: pointing to endpoint of the left-most 1st bin-group **Right_Pointer**: pointing to the end point of the right-most bin-group **Span**: the difference between the right-most and the left-most bins **Total_area**: summation of all bins **Dist_right**: the distance between the Left_Pointer and its right-neighboring bin-group **Dist_left**: the distance between Right_Pointer and its left-neighboring bin-group **Area_right**: the area of the Left_Pointer's right-neighboring bin-group **Area_left**: the area of Right_Pointer's left-neighboring bin-group

本論文で提案する学習アルゴリズムの概要は**表 3**（初期化部分）と**図 5**（主体部分）に示すとおりである．すべての bin はいくつかの zero-bin によって分割されていると仮定する．ただ 1 つの bin-group しか存在しない場合は，このような bin-group をすべて正常トラフィックと仮定し，この bin-group の右端を学習結果とする．bin-group の面積はその bin-group に含まれる bin の度数（高さ）の合計を意味する．

4.2 概要

本論文で提案する学習アルゴリズムは，Right_Pointer と Left_Pointer という 2 つのポインタを利用する．最初，Right_Pointer は一番右側の bin-group の右端に置き，異常な bin-group を調べる目的で利用する．一方 Left_Pointer は一番左側の bin-group の右端に置き，通常な bin-group を調べる目的で利用する．Left_Pointer はある条件（条件 1）によって左から右へ bin-group ごとに進む．Right_Pointer は別の条件（条件 2）によって右から左へ bin-group ごとに進む．この 2 つのポインタとその移動条件を導入することにより，本研究では学習アルゴリズムの自動化を実現することができた．条件 1 と条件 2 の詳細は 4.3 節で説明する．この 2 つのポインタは一致するまで繰り返して移動さ

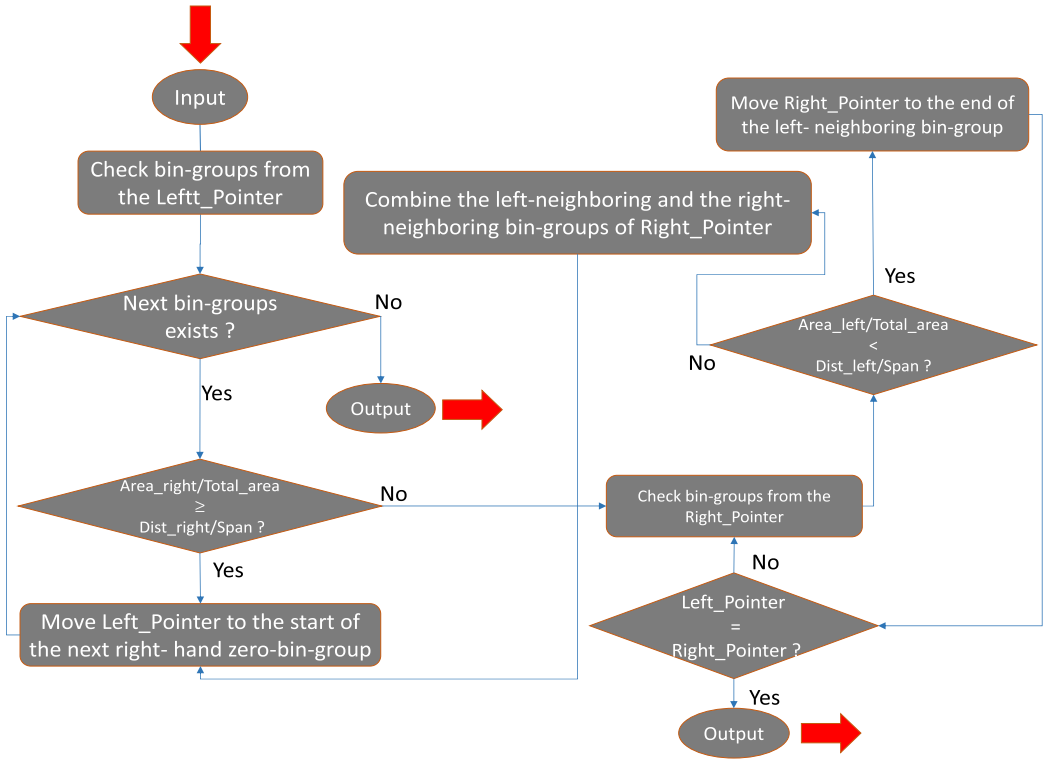


図 5 本研究で提案した学習アルゴリズム
Fig. 5 Learning algorithm proposed in this study.

せ，一致したところを学習結果とする．もし 2 つのポインタが一致する前に，両方も止まったら，4.4 節にて導入する bin-group の合併を行う．bin-group の合併を用いることで，4.5 節で示すように 2 つのポインタが必ず一致にすることを保証できる．

4.3 Right_Pointer と Left_Pointer の移動条件

Right_Pointer の移動条件は次のとおりである．

条件 1：

$$\frac{Dist\_left}{Span} > \frac{Area\_left}{Total\_area}$$

ここで Dist_left は Right_Pointer とその左に隣接する bin-group との距離であり，Span は横軸にある最左端の bin と最右端の bin との距離である．Area_left は Right_Pointer の左に隣接する bin-group の面積であり，Total_area はすべての bin の度数（縦軸の値）の和である．この条件は Right_Pointer の左側にある bin-group は異常か否かを判別するために，その bin-group の面積割合と，その bin-group からその bin-group の左に隣接する bin-group までの距離割合を用いる．面積の割合は小さいほど，かつ距離の割合が大きいほど，異常と判別される可能性が高くなる．

Left_Pointer の移動条件は次のとおりである．

条件 2：

$$\frac{Dist\_right}{Span} \leq \frac{Area\_right}{Total\_area}$$

ここで Dist_right は Left_Pointer とその右に隣接する bin-group との距離であり，Area_ right は Left_Pointer の右に隣接する bin-group の面積である．この条件は，Left_Pointer の右側の bin-group が通常か否かを判別するために，その bin-group の面積割合と，その bin-group とその bin-group の右に隣接する bin-group の距離割合を用いる．面積の割合は大きいほど，かつ距離の割合が小さいほど，通常と判別される可能性が高くなる．

4.4 終了またはリピート

2 つのポインタが一致した場合，その地点を学習結果として採用する．一方，2 つのポインタが一致する前に止まった場合，Right_Pointer 左側にある 2 つの bin-group を合併させる．その後，再び 2 つのポインタの移動条件を適用し移動を繰り返す．この学習アルゴリズムは必ず 2 つのポインタが一致して終了することを 4.5 節で説明する．

4.5 収束

図 5 で示すように，bin-group の合併を行ったあと，この学習アルゴリズムはポインタの移動を繰り返して実行する．bin-group の合併が発生する最悪の場合を考えると，このとき，2 つのポインタの間に 1 つの bin-group しか存在しない．このとき，Dist_left と Dist_right，Area_left と Area_right は等しくなる．したがって，Right_Pointer また Left_Pointer のどちらは必ず移動する．なぜなら，2 つのポインタの移動条件は逆になっているからである．その結

果，2 つのポインタは一致してアルゴリズムは終了する．

以上のように，パラメータが要らない自動学習アルゴリズムを実現することができる．なお，度数分布のパラメータは 3 章で紹介した既存の研究 FHST 手法 [5] と同じである．

5. 実験

提案手法では，学習アルゴリズムを適用する前に度数分布を作成する必要がある．そのためには，時間単位と bin 幅を決める必要がある．この章では，まず，同じデータセットに対して作成した異なる度数分布がどのように学習の結果に影響するかを調査する．そして，学習アルゴリズムの検知性能を検証する．

まず 5.1 節では度数分布図の学習結果への影響を検証する．本論文では，データ量の多いダークネットから収集されたデータを用いる．このデータは異常データを明示するラベル付きの ground-truth データがないため，本提案の検知結果を評価することは難しい．そこで，5.2 節では，ラベル付きの異常データを含む人工データを用いて，本提案の検知性能を検証する．すなわち，5.1 節と 5.2 節で利用するデータセットは異なる．

5.1 度数分布図の学習結果への影響

提案アルゴリズムを適用する度数分布図を作成するためには，時間単位と bin 幅を決める必要がある．時間単位はトラフィックを集計する最小時間幅として，異常に対する検知の速さと直接に関連する．具体的にいえば，時間単位が長くなると異常検知は遅れ，短すぎると通常モードが少なくなり検知システムは異常に対する過敏症状が出る．適切な bin 幅は，必要な通常モードの抽出精度によって変わる．本節では，時間単位と bin 幅が及ぼす学習結果への影響を調べる．なお，時間単位と bin 幅は度数分布を作成する際のパラメータであるため，本提案の学習アルゴリズムのパラメータではない．

5.1.1 実験データ

一般的に，実際のトラフィックデータを収集することは難しい．一方，多くの研究が，ダークネットデータの有効性を示している [17], [18], [19], [20], [21]．ダークネットとはインターネット上にある未使用な IP アドレスからなるネットワークである [22]．このダークネットデータを使ったスキャン検知の研究は数多く存在している．そこで，我々の実験もダークネットデータを使って提案アルゴリズムの性能を検証する．今回の実験は独立行政法人情報通信研究機構（NICT）から提供されたダークネット観測データ（2011 年 6 月，30 日間約 8,799 万 TCP パケット）を利用する．

本実験では，時間単位として 10 分，30 分，60 分の 3 種類を，bin 幅として 250，500，750，1,000，2,000，3,000 の 6 種類を設定した．実験結果によると bin 幅は実験結果に大きく影響を与えていない．このため，ここでは 500 と 1,000 を bin 幅の例としてあげる．

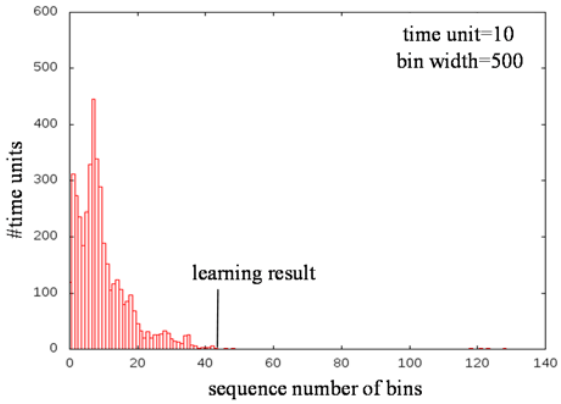


図 6 度数分布図：時間単位=10 分，ビン幅=500
Fig. 6 Frequency distribution: time unit=10, bin width=500.

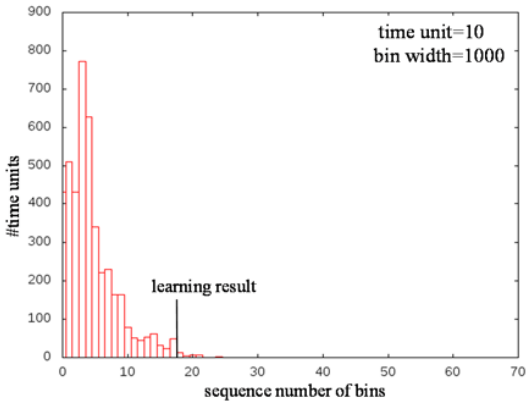


図 7 度数分布図：時間単位=10 分，bin 幅=1,000
Fig. 7 Frequency distribution: time unit=10, bin width=1,000.

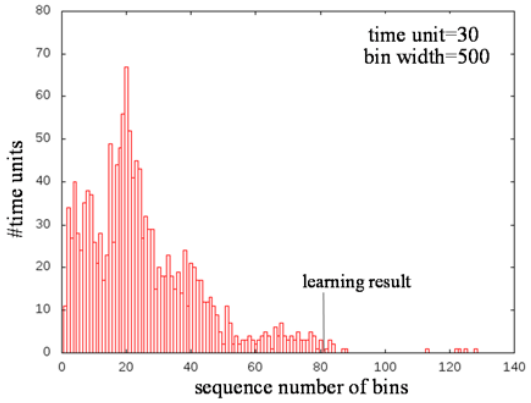


図 8 度数分布図：時間単位=30 分，bin 幅=500
Fig. 8 Frequency distribution: time unit=30, bin width=500.

5.1.2 実験結果

図 6，**図 7**，**図 8**，**図 9**，**図 10**，**図 11** はそれぞれの条件において得られた度数分布図である．図 6 によると，時間単位を 10 分間隔とし，bin 幅を 500 とした場合，学習結果は 44 番目の bin のところとなっている．すなわち，学習結果として，通常モードは 44 × 500（bin 幅）= 22,000

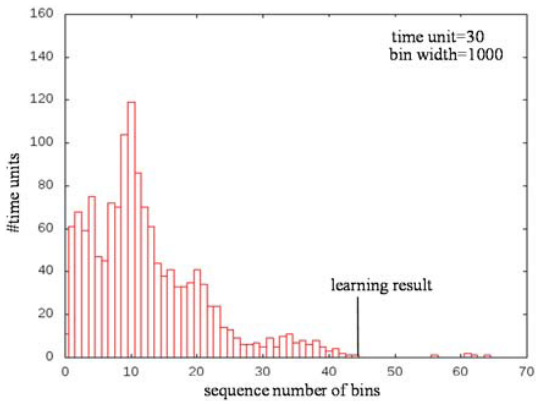


図 9 度数分布図：時間単位=30 分，bin 幅=1,000
Fig. 9 Frequency distribution: time unit=30, bin width=1,000.

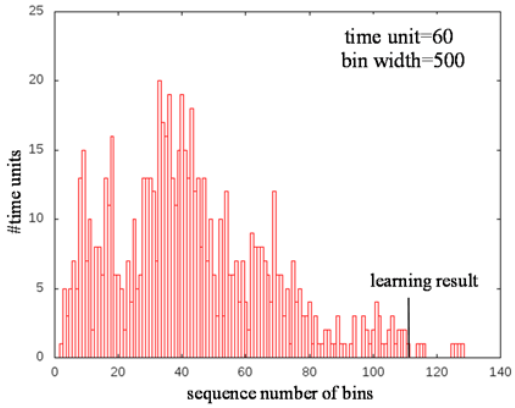


図 10 度数分布図：時間単位=60 分，bin 幅=500
Fig. 10 Frequency distribution: time unit=60, bin width=500.

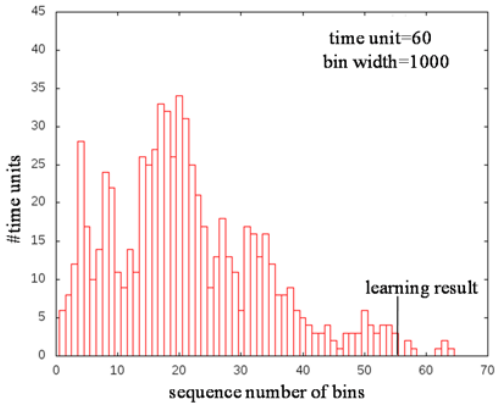


図 11 度数分布図：時間単位=60 分，bin 幅=1,000
Fig. 11 Frequency distribution: time unit=60, bin width=1,000.

になる．図 7 は，時間単位は同じく 10 分間隔で，bin 幅を 1,000 とした場合である．このとき，通常時モードは 22 × 1,000（bin 幅）= 22,000 となっていた．図 6 と図 7 では，時間単位が比較的小さいため度数分布図全体が左に寄っていることが分かる．

図 8 は，時間単位として 30 分間隔を選び，bin 幅を 500 とした場合である．このとき，通常モードは 81×500 = 40,500 になる．図 9 は，時間単位として同じく 30 分間隔で，bin 幅として 1,000 を選んだ場合である．このとき，通常モードは 45 × 1,000 = 45,000 になる．図 8 と図 9 によると，時間単位が大きくなると，その時間単位内で観測されるアクセス数が全体として増えるため，学習結果も大きくなる．一方，同じ時間単位に対して bin 幅を変えると，学習結果が変化することがある．図 10 は，時間単位として 60 分間隔を選び，bin 幅として 500 を選んだ場合である．このとき，通常モードは 112 × 500 = 56,000 になる．図 11 は，時間単位は同じく 60 分間隔で，bin 幅として 1,000 を選んだ場合である．このとき，通常モードは 56 × 1,000 = 56,000 である．図 10 と図 11 によると，時間単位が大きすぎると度数分布図は全体的に右へ寄る傾向がある．

5.1.3 考察

図 6〜図 11 の実験結果により，次の考察が得られる．

a) 時間単位が増えると，通常モードの範囲は大きくなる傾向がある．その原因は時間単位内にアクセスされたポート数は大きくなるからである．

b) ある時間単位において，bin 幅を変えても学習結果に大きな影響はない．しかしながら，図 8 と図 9 に示した結果は異なる．その原因は，bin 幅が増えると，bin-group の合併により，bin の分布が変わったからである．

5.2 異常トラフィックの学習結果への影響

5.2.1 データ

異常トラフィックの検知実験に用いたデータは，当研究室のサーバで収集されたものに攻撃データを追加した人工データである．研究室ウェブページへの通信やメールなどはこのサーバを経由しているため，日常的なトラフィックが記録されている．本実験ではこのデータを正常なトラフィックとして扱う．

一方，異常データすなわち攻撃データを作るために，Nmap [24] を利用した．Nmap はポートスキャンや OS 検出など多くの機能を兼ね備えているソフトウェアであり，よく利用されているセキュリティスキャナである．

1) 学習データ

本実験では正常なトラフィックとして当研究室サーバ取集した 2 日間（2014 年 11 月 8 日〜2014 年 11 月 9 日）のトラフィックデータ（48 時間約 500 MB の TCP パケット）を利用した．この通常トラフィックに異常トラフィックとして，Nmap を使って下記の 3 パターンのポートスキャン攻撃によって得たトラフィックを混ぜた．それぞれの学習結果を分析することにより，異なるポートスキャンデータの学習結果への影響を調べる．

パターン 1 は，intense scan と呼ばれるポートスキャン攻撃である．この種の攻撃は，危険性が高いポートスキャンとして，よく使われる手段の 1 つである [24]．パターン

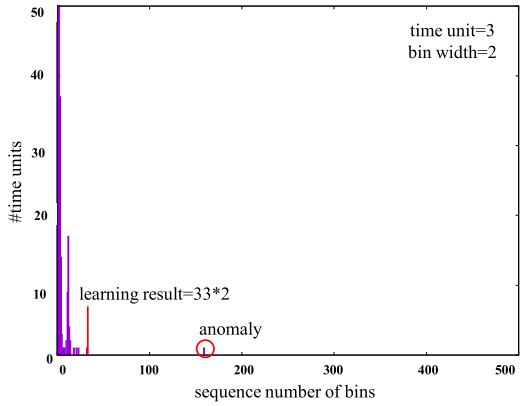


図 12 度数分布図および学習結果（パターン 1）
Fig. 12 Frequency distribution and learning result (pattern 1).

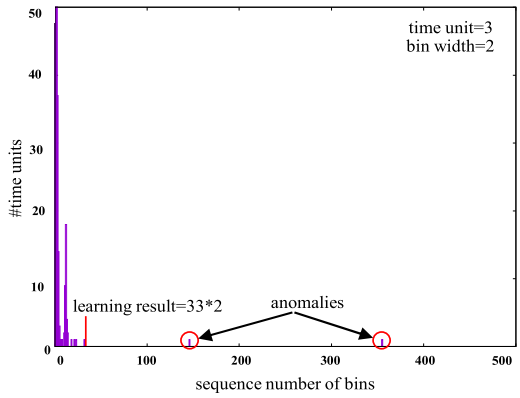


図 13 度数分布図および学習結果（パターン 2）
Fig. 13 Frequency distribution and learning result (pattern 2).

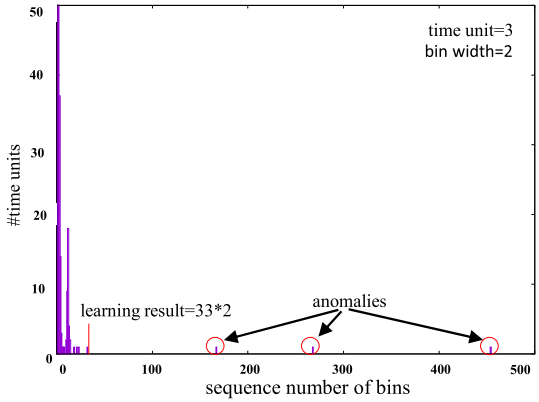


図 14 度数分布図および学習結果（パターン 3）
Fig. 14 Frequency distribution and learning result (pattern 3).

2 は，2 種類のポートスキャン攻撃を含んでいる．1 つは intense scan で，もう 1 つはより高速のポートスキャンである．パターン 3 では，3 種類のポートスキャン攻撃を行った．Intense scan 以外に，スキャン速度（単位時間内にスキャンしたポート数）が異なる 2 種類のポートスキャンを加えたものである．なお，同じ種類の攻撃であってもランダム性およびパラメータにより同じ攻撃データになるとは限らない．

2) テストデータ

テストデータは 2 日間（2014 年 11 月 12 日～2014 年 11 月 13 日）に各 1 回のスキャン攻撃を行って収集したトラフィックデータである．データの流れを**図 15** に示す．

5.2.2 度数分布の作成

前述したように，学習アルゴリズムを使う前に度数分布を作成する．5.1 節では時間単位と bin 幅は及ぼす度数分布図への影響を論じた．今回の実験は研究室のサーバで収集した小規模のデータであるため，時間単位と bin 幅を小さい数値に設定すれば提案の学習アルゴリズムの検知性能を実証することができると考えられる．今回の実験で時間単位は 3 分間隔とし，bin 幅は 2 とした．48 時間のデータであるため，960 時間単位がある．

5.2.3 学習結果

上述した 3 つのパターンで作成した度数分布および学習の結果を**図 12**，**図 13** および**図 14** に示す．学習結果を示す bin は 3 つのパターンとも同じく 33 番 bin で，通常モードは 33×22（bin 幅）$= 66$ となった．

以上の実験結果により，今回の例では異なる種類の異常通信が含まれても学習結果が変わらないことが分かる．

5.2.4 関連研究と学習結果の比較

3 章で挙動に基づく検知手法を利用した関連研究の FHST 手法 [5] を紹介した．本項では図 12～図 14 の人工データを使って FHST 手法との学習結果の比較を行う．FHST アルゴリズムを使うにはパラメータを決める必要があるため，本論文では 9 組のパラメータを試した．比較結果は**表 4** に示す．この比較の結果からは次のことが分かる．

表 4 FHST 案との学習結果の比較
Table 4 Comparison on learning result.

	Parameters (α, β)	図 12 データ	図 13 データ	図 14 データ
FHST 手法	(0.10, 0.30) (0.18, 0.30) (0.20, 0.30) (0.20, 0.20) (0.20, 0.40) (0.20, 0.33) (0.20, 0.27) (0.22, 0.30)	66	66	66
	(0.30, 0.30)	66	66	538
本論文提案	None	66	66	66

1) FHST 手法では，9 組のパラメータの中，8 組は同じ学習結果を得たため，パラメータは学習結果に影響し

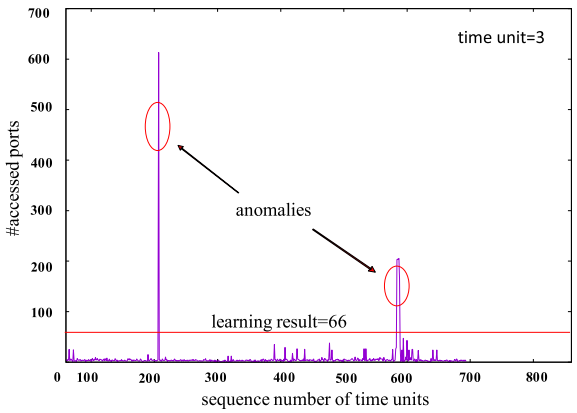


図 15 テストデータ：通信トラフィック時系列
Fig. 15 Test data: traffic data including anomalies.

ていないように見える．しかし，これはデータセットに依存する．

2) FHST 手法では学習に失敗することがあった．本実験では，パラメータが $\alpha = 0.3$，$\beta = 0.3$ のとき，FHST 手法の学習結果を示す bin は 269 番の bin になった．bin 幅が 2 であることから通常モードは 538 である．すなわち，図 14 における左から 1 番目と 2 番目の異常トラフィックグループを通常と判断してしまっていた．なぜなら，図 14 にある 3 つの異常の中，左側 2 つの異常間の距離は，横軸全体の 20%より大きいが 30%より小さい．そのため，α を 0.2 とするか 0.3 とするかによって結果は大きく変わった．この学習結果を通常モードとして利用すれば，明らかに異常の検知漏れが多く生じることになる．

3) 本論文での提案手法は，パラメータチューニングを行うことなく FHST 手法の最良結果と同じ結果を得た．すなわち，2) で述べたような FHST 手法の失敗例を回避できた．したがって，本提案は学習性能を犠牲せず学習の自動化を実現したといえる．

5.2.5 学習アルゴリズムの実行時間

パターン 3 の実験（図 14）に用いたデータ約 500 MB を利用して本提案手法と FHST 手法の実行時間を比較した．両手法は度数分布図の作成まではまったく同じプロセスで，学習アルゴリズムのみが異なっている．実験には，Core i7 4,960k の CPU と 8 GB のメモリを持つサーバ上で動作する仮想マシンを利用した．度数分布図の作成を含めて学習の結果が得るまで，両手法とも 10 秒弱の時間を要した．学習段階のみを見れば，両手法の所要時間は FHST 手法 0.6 ms に対して，本提案手法 0.9 ms であった．これにより，学習段階の所要時間は，両手法が同じである度数分布の作成時間に比べて無視できるほどである．

データ量が大きくなっても，学習アルゴリズムの実行時間は Bin の数と Bin の分布にのみ依存するため，学習トラフィックデータのサイズには依存しない．なお，度数分布の作成には，すべてのデータを 1 回走査必要があるため，全体では実行時間はデータ量 n に対して $O(n)$ の関係にある．

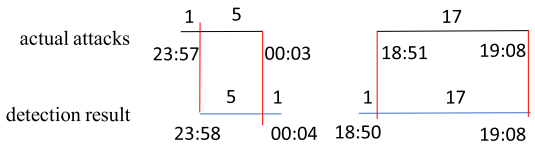


図 16 実際の攻撃と検知の結果
Fig. 16 Actual anomalies and detection result.

表 5 検知結果の評価
Table 5 Evaluation on detection result.

	検知率	誤検知率	見逃し率
提案手法	$\frac{5+17}{6+17}=95.7\%$	$\frac{1+1}{6+18}=8.3\%$	$\frac{1}{6+17}=4.3\%$
FHST 手法	同上	同上	同上

5.3 ポートスキャン攻撃の検知

これまでの実験では，トラフィックデータの変化と学習結果への影響を調べた．本節では，学習で得られた通常モード 66（表 4 を参照）を図 15 で示したテストデータに含まれる異常トラフィックを検知してみることにより，本論文で提案した学習アルゴリズムの有効性を検証する．本実験では 1 分間を最小単位とし，検知率などを計る．検知結果は**図 16** と**表 5** で示す．

実験結果によれば，提案の学習アルゴリズムによる学習結果をポートスキャン検知に用いれば，95%以上の検知率を達成した．なお，FHST 手法の学習結果は最良のパラメータを選択した場合，本提案の結果と同じであるため，検知性能も同じである．したがって，パラメータチューニングが必要ない本論文の提案手法も同等の良い検知結果を得たといえる．

5.4 データ規模を拡大した実験

本節では，通常トラフィックデータおよび攻撃トラフィックデータの規模を拡大して本提案の学習アルゴリズムの動作を確認する．まず，今まで使った 2 日間の通常データを 15 日間に拡大するとともに，図 14 で示した攻撃パターン 3 の攻撃データを 10 倍にする．作成した度数分布図および学習結果を**図 17** に示す．具体的には，図 14 と同じ学習結果を得た．したがって，検知性能も同じであった．

そして，データ量が増えたことにより，度数分布図で異常トラフィックの分布を正常トラフィックの分布に近づけさせて提案手法の性能を考察する．作成した度数分布およ

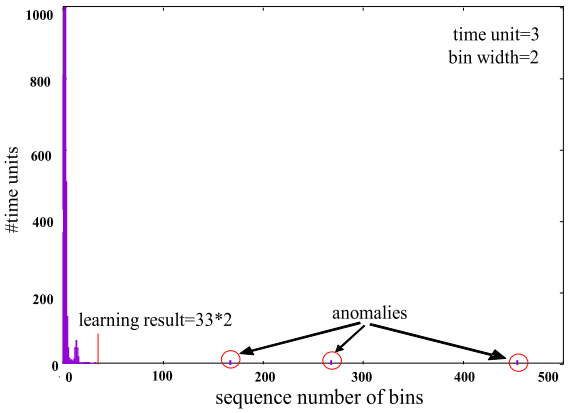


図 17 拡大したデータの度数分布と学習結果（ケース 1）
Fig. 17 Enlarged learning data (case 1).

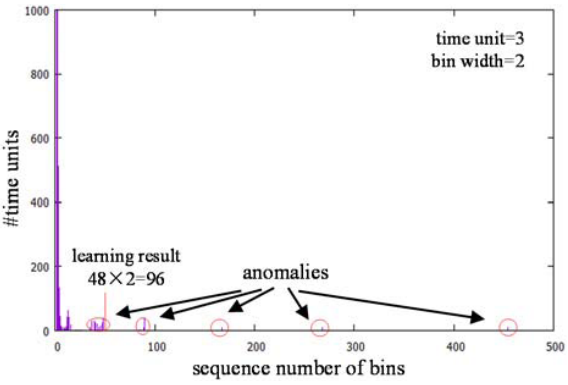


図 18 拡大したデータの度数分布と学習結果（ケース 2）
Fig. 18 Enlarged learning data (case 2).

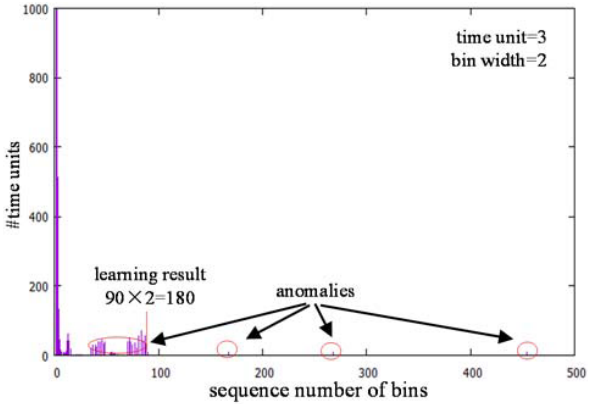


図 19 拡大したデータの度数分布と学習結果（ケース 3）
Fig. 19 Enlarged learning data (case 3).

び学習結果を**図 18** に示す．学習結果は図 17 の学習結果より大きくなった．すなわち，学習アルゴリズムは異常トラフィックの一部，スローポートスキャンといわれる攻撃部分を通常トラフィックと判断した．しかしながら，表 5 とまったく同じ検知結果を得た．すなわち，学習で得た通常モードが多少変化しても検知結果に影響を及ぼすには限らない．

最後に，スローポートスキャンのスピード範囲がもっと広がった場合を調べる．**図 19** は一例である．このケースでは，スローポートスキャンはスキャンスピードを変更しながら多い回数で攻撃した．このような場合は学習結果には大きな影響があり，学習結果は 180 に上がった．この学習結果を利用する検知結果として，図 15 で示したテストデータの中にある 2 回合わせて 23 分間の攻撃は 17 分間検知できた．すなわち，検知率は 73.9%に落ちた．すなわち，スキャンスピードを変えながら行った大量のスローポートスキャン攻撃トラフィックを含むデータを学習データとして利用すると，提案アルゴリズムはそのスローポートスキャンに訓練させられ，検知性能が落ちた．

5.5 考察

実験結果から次のことが分かった．

1) 本研究で提案したパラメータなしの学習アルゴリズムは 2 つのパラメータを必要とする既存の FHST 手法による最良の結果と同等の検知性能を得た．すなわち，本研究では有効な自動学習アルゴリズムを実現することができた．
2) 本論文における提案手法は，スキャンスピードを変えながら行った大量のスロースポートスキャンに訓練させられることがあり（図 19 のケースは一例），その場合は検知性能が落ちる可能性がある．

6. 結論と今後の課題

本論文では挙動に基づく検知手法に対して，学習アルゴリズムの重要性を論じた．しかし，既存の一般的な学習アルゴリズムはパラメータが必要であり，そのパラメータを事前に決めるのが難しいという問題点がある．本研究では 2 つのポインタを利用することにより，パラメータのチューニングが省かれた学習アルゴリズムを提案した．実験の結果により，提案学習アルゴリズムの有効性を示した．

今後の課題として，データの量および攻撃の種類をより多く収集して提案した学習アルゴリズムの性能をさらなる実証する．

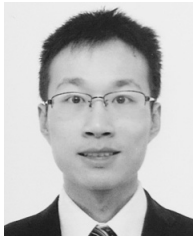
謝辞 この研究の一部は，総務省による「国際連携によるサイバー攻撃の予知技術の研究開発」および科学研究費（基盤研究（C）No.25330131）の支援を受けている．

本研究を実施するにあたり，独立行政法人情報通信研究機構（NICT）よりダークネット観測データの提供を受けた．ここに記して謝意を表す．

参考文献

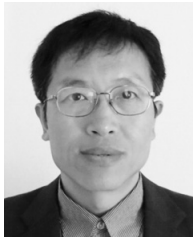
[1] 総務省：平成 26 年版情報通信白書，総務省（オンライン），入手先 ⟨http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h26/html/nc253120.html⟩（参照 2014-12-05）．
[2] 山田正平，見神宏紀，木村啓二ほか：不正侵入検知システムにおけるマルチコア上でのシグネチャ割当によるレイテンシ削減手法，情報処理学会研究報告，Vol.2014-ARC-209,

No.2, pp.1–8 (2012).
[3] Modia, C., Patela, D., Borisaniya, B., et al.: A survey of intrusion detection techniques in Cloud, *Journal of Network and Computer Applications*, Vol.36, No.1, pp.42–57 (2013).
[4] Denning, D.E.: An Intrusion-Detection Model, *IEEE Trans. Software Engineering - Special Issue on Computer Security and Privacy*, Vol.13, No.2, pp.222–232 (1987).
[5] Feng, Y., Hori, Y., Sakurai, K., et al.: A Behavior-Based Method for Detecting Distributed Scan Attacks in Darknets, *Journal of Information Processing*, Vol.21, No.3, pp.527–538 (2013).
[6] 情報処理推進機構：セキュリティ担当者のための脆弱性対応ガイド，情報処理推進機構（オンライン），入手先 ⟨http://www.ipa.go.jp/files/000024184.pdf⟩（参照 2014-12-05）.
[7] 情報処理推進機構：ポートスキャン，情報処理推進機構（オンライン），入手先 ⟨https://www.ipa.go.jp/security/fy14/contents/soho/html/chap1/scan.html⟩（参照 2014-12-05）.
[8] Dabbagh, M., Ghandour, A., Fawaz, K., et al.: Slow Port Scanning Detection, *Proc. 7th International Conference on Information Assurance and Security* (*IAS2011*), pp.228–344 (2011).
[9] Aniello, L., Lodi, G. and Baldoni, R.: Inter-Domain Stealthy Port Scan Detection through Complex Event Processing, *Proc. 13th European Workshop on Dependable Computing*, pp.67–72 (2011).
[10] Chowdhary, M., Suri, S. and Bhutani, M.: Comparative Study of Intrusion Detection System, *International Journal of Computer Sciences and Engineering* (*JCSE*), Vol.2, No.4, pp.197–200 (2014).
[11] Gadge, J. and Patil, A.A.: Port scan detection, *Proc. 16th IEEE International Conference on Networks* (*ICON 2008*), pp.1–6 (2008).
[12] Bhuyan, M.H., Bhattacharyya, D.K. and Kalita, J.K.: Surveying Port Scans and Their Detection Methodologies, *The Computer Journal*, Vol.54, No.10, pp.1565–1581 (2011).
[13] Treurniet, J.: A Network Activity Classification Schema and Its Application to Scan Detection, *IEEE/ACM TON*, Vol.19, No.5, pp.1396–1404 (2011).
[14] Ensafi, R., Park, J.C., Kapur, D., et al.: Idle Port Scanning and Non-interference Analysis of Network Protocol Stacks Using Model Checking, *Proc. 19th USENIX Security Symposium* (2010).
[15] Zhang, Y. and Fang, B.: A Novel Approach to Scan Detection on the Backbone, *Proc. ITNG'09*, pp.16—21 (2009).
[16] Kong, S., He, T., Shao, X., et al.: Scalable Double Filter Structure for Port Scan Detection. *Proc. ICC'06*, pp.2177–2182 (2006).
[17] Jung, J., Paxson, V., Berger, A.W., et al.: Fast Portscan Detection Using Sequential Hypothesis Testing. *Proc. SECPRI'04*, pp.211–225 (2004).
[18] Akimoto, S., Hori, Y. and Sakurai, K.: Collaborative Behavior Visualization and its Detection by Observing Darknet Traffic, *Proc. 4th International Symposium on Cyberspace Safety and Security* (*CSS*), LNCS 7672, pp.212–226 (2012).
[19] Eto, M., Inoue, D., Song, J., Ohtaka, K., et al.: Nicter: A Large-Scale Network Incident Analysis System, *Proc. the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security* (*BADGERS*), pp.37–45 (2011).
[20] Bailey, M., Cooke, E., Jahanian, F., et al.: The Internet Motion Sensor: A distributed blackhole monitoring system, *Proc. 12th ISOC Symposium on Network and Distributed Systems Security* (*NDSS*), pp.167–179 (2005).
[21] Pouget, F., Dacier, M. and Pham, V.H.: Leurre.com: On the Advantages of Deploying a Large Scale Distributed Honeypot Platform, *Proc. E-Crime and Computer Conference* (*ECCE*) (2005).
[22] Liston, K.: Fun with Darknets, SANS Internet Storm Center, available from ⟨http://isc.sans.org⟩ (accessed 2014-12-05).
[23] Cooke, E., Bailey, M., Mao, Z.M., et al.: Toward Understanding Distributed Blackhole Placement, *Proc. ACM CCS Workshop on Rapid Malcode*, pp.54–64 (2004).
[24] Wikipedia: nmap, wikipedia (online), available from ⟨http://ja.wikipedia.org/wiki/Nmap⟩ (accessed 2014-12-05).
[25] Wang, C., Feng, Y. and Kawamoto, J., et al.: A Parameterless Learning Algorithm for Behavior-Based Detection, *Proc. 9th Asia Joint Conference on Information Security* (*AsiaJCIS2014*) (2014).



王 サン

2012 年（中国）南京航空宇宙大学情報工学専攻卒業．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）．同年より東日本電信電話株式会社に勤務．現在に至る．



フォン ヤオカイ （正会員）

1986 年（中国）天津大学計算機科学科卒業．1992 年同大学大学院計算機科学研究科修士課程修了．（中国）天津大学計算機科学系助教・講師を経て 1998 年九州大学大学院システム情報科学研究科に留学．九州大学博士（情報学，2004 年）．現在，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情報セキュリティ研究室特別研究員（兼務）．ネットワークセキュリティ，パターン認識，データベースの研究に従事．IEEE，日本データベース学会各会員．

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．