自己組織化マップによるネットワークトラフィックの逐次学習と異常検出

二瓶 凌輔[1,a)]　青木 茂樹[1,b)]　宮本 貴朗[1]

概要：近年，サイバー攻撃が増加しており，サイバー攻撃を検出するための研究が盛んに行われている．サイバー攻撃は常に変化しており，攻撃を持続的に検出するためには，攻撃の変化に逐次対応させる必要がある．そこで本稿では，逐次学習と異常検出を行える機械学習手法を用いることで，サイバー攻撃が変化する場合にも持続的に攻撃を検出できる手法を提案する．一般に，機械学習では学習を行う際に教師ラベルが必要となる手法が多い．本手法では，特徴ベクトルのマハラノビス距離に基づいて外れ値を検出し，教師ラベルとして利用する．まず，トラフィックデータからパケットのサイズなどの特徴量を抽出し，特徴量ごとにエントロピーの算出を行い特徴ベクトルを生成する．抽出した特徴ベクトルと教師ラベルを用いて，自己組織化マップで逐次学習と異常検出を行う．本手法ではトラフィックを逐次学習して異常検出するため，サイバー攻撃の変化に対応し，ネットワークの異常を検出することができる．実験では，MWSデータセットを用いて，本手法の有効性を確認した．

キーワード：ネットワーク異常検出，逐次学習，自己組織化マップ

Online Learning and Anomaly Detection of Network Traffic using Self-Organizing Maps

Ryosuke Nihei[1,a)]　Shigeki Aoki[1,b)]　Takao Miyamoto[1]

Abstract: In recent years, cyber attacks are increasing and researches for detecting cyber attacks have been actively studied. Cyber attacks are constantly changing. In order to detect the attacks continuously, it is necessary to respond to change in the attacks sequentially. In this paper, we propose a method that can detect the attacks continuously even when the cyber attacks change by using a machine learning method that can perform sequential learning and anomaly detection. In general, in machine learning, there are many methods that require supervised data for learning. In our method, outliers are detected based on the Mahalanobis distance of feature vectors extracted from network traffic data and used as supervised data. First,we extract feature quantities such as packet size from traffic data and calculate entropy for each feature quantity, and generate feature vectors. By using the feature vectors and the supervised data extracted by the above outlier detection, we perform sequential learning and anomaly detection using a self-organizing map. And it is possible to detect network anomalies in response to change in the cyber attacks. In the experiment, we confirmed the effectiveness of our method using MWS dataset.

Keywords: Anomaly Detection of Network, Online Learning, Self-Organizing Maps

1. はじめに

近年，インターネットの普及に伴ってサイバー攻撃による被害が後を絶たず，社会的な問題となっている．サイバー攻撃を検出する手法として，ネットワーク上の不正な

[1] 大阪府立大学大学院人間社会システム科学研究科
Graduate School of Humanities and Sustainable System Sciences, Osaka Prefecture University
[a)] sza04214@edu.osakafu-u.ac.jp
[b)] aoki@kis.osakafu-u.ac.jp

トラフィックを検出する侵入検知システム (IDS:Intrusion Detection System) の研究が盛んに行われている．IDS はシグネチャ型とアノマリ型の 2 種類に大別できる．代表的なシグネチャ型 IDS には，Snort[1] や Suricata[2]，The Bro[3] 等が挙げられる．シグネチャ型 IDS は異常を定義したパターンファイルに基づいて異常の検出を行う方式である．この方式には，パターンファイルに定義した攻撃は検出できるが，定義されていない攻撃を検出できない欠点がある．文献 [4] では，シグネチャ型 IDS の検出結果から学習データを自動生成し，機械学習を適用することで本来検出できない未知の攻撃を検知できる IDS を提案している．しかしこの手法では，パターンファイルに基づいて学習データを生成しているために，パターンファイルに登録されていない新規の異常を検出できないことが課題となっている．

一方, 代表的なアノマリ型 IDS としては文献 [5], [6], [7], [8] の手法が挙げられる．アノマリ型 IDS は正常な通信のみを含むデータを学習し，そこから外れた状態を異常として検出する方式である．アノマリ型 IDS を，シグネチャ型 IDS と比較すると，パターンファイルに定義されていない未知の攻撃を検出できるが，誤検知が多いことが欠点である．

これまでに提案されている IDS では，事前に定義した異常や学習した結果に基づいた異常は検出できるもののサイバー攻撃の変化に対応した持続的な攻撃の検出については考慮されていなかった．サイバー攻撃は常に変化しており，攻撃を持続的に検出するためには攻撃の変化に逐次対応する必要があると考えられる．そこで文献 [9], [10] では，逐次的な学習が可能な自己組織化マップ（Self-Organizing Maps，以後 SOM とする）を用いた異常検出手法を提案している．文献 [9] では，プログラムを動作させた際に呼び出されるシステムサービスの情報から文字列を抽出し，生成された特徴ベクトルで SOM を学習し，マルウェアの感染を検出する手法を提案している．また文献 [10] では，パケットのヘッダから抽出した特徴ベクトルに決定木を適用して教師ラベルを付与し，特徴ベクトルと付与した教師ラベルを SOM で逐次学習しながら異常を検出する手法を提案している．

本稿では，パケットのヘッダから抽出した特徴量のエントロピーを求めて特徴ベクトルとし，そのマハラノビス距離を基に検出した外れ値により教師ラベルを作成して SOM で逐次学習しながら異常検出する手法を提案する．実験では，MWS データセットに本手法を適用して有効性を確認した．

2. 関連研究

本研究に関連する従来研究として，アノマリ型 IDS の代表的な手法である文献 [5], [6], [7], [8] と自己組織化マップを用いて異常を検出する手法である文献 [9], [10] について述べる．

文献 [5] では，パケットのヘッダ情報から得られたエントロピーにより異常を検出する手法を提案している．この手法ではパケットを到達した順番に並べ，一定のパケット数で分割する．そして，分割したパケットから IP アドレスやポート番号など毎にパケット数を計測する．次に，パケットの出現確率を求め，求めた出現確率からエントロピーを算出する．その後，EMMM 法によりエントロピーの時系列変化に着目して，エントロピーの変化が大きい時間を攻撃などが含まれている状態として検出している．

文献 [6] では，ネットワークのトラフィックは複数の正常状態で表されると考え，定義された複数の正常状態との違いから異常を検出する手法を提案している．この手法では，異常を含まないデータから単位時間当たりの ICMP や TCP パケット数等を計測してクラスタリングする．メンバが少ないクラスは削除し全てのクラスにおいて閾値以上のメンバ数となるまでクラスタリングを繰り返す．クラスタリング結果を正常状態として定義し，新たに観測されたデータから同様の特徴を抽出し，正常クラスとの距離が閾値以上かどうかで異常の判別を行っている．

文献 [7] では，複数の特徴量の組み合わせによる異常検出手法を提案している．この手法では，異常をトラフィック量の異常，通信手順の異常，通信内容の異常の 3 種類に分け，単位時間あたりのトラフィック量を数値化した特徴量，フロー毎のフラグの出現回数を数値化した特徴量，フロー内のパケットのペイロードのパターンの傾向を数値化した特徴量を学習用データからそれぞれ抽出する．そして新たなデータでこれらの特徴量を抽出し，学習用データの値と閾値以上離れている特徴量が存在する場合に異常であると判断する．

文献 [8] では，パケットトレースから一般的な通信の特徴量 70 種類とマルウェア特有の特徴量 25 種類を抽出し，DBSCAN を用いてクラスタリングする．異常検出時にはパケットトレースから 95 種類の特徴量を抽出し，クラスタ内のデータとのユークリッド距離を算出する．算出した距離の最小値が閾値未満の場合は既存のクラスタに分類され既知のマルウェアとし，閾値以上の場合は未知のマルウェアとしている．

既存の IDS では，事前に定義した異常や学習した結果に基づいた異常は検出できていた．しかし，サイバー攻撃の変化に対応した持続的な攻撃の検出については考慮されていない．攻撃を持続的に検出するためには，攻撃の変化に逐次対応させる必要がある．そこで，文献 [9], [10] では逐次学習を行える SOM を用いた手法を提案している．

文献 [9] では，プログラムを動作させた際に呼び出される Windows システムサービスの呼び出し列と入出力パラメータに設定される文字列を用いてマルウェアを検出する手法を提案している．正常なデータと異常なデータから文

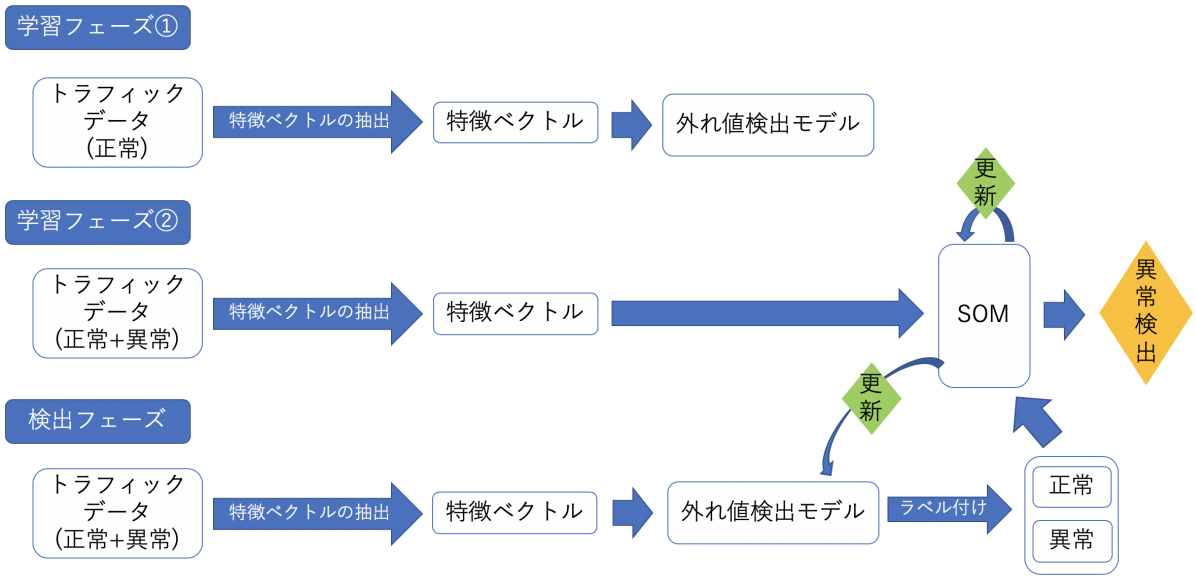

図 1　提案手法の概要
Fig. 1　Outline of Proposed Method.

字列を抽出し，生成した特徴ベクトルを SOM の入力データとしてマップを作成する．マップ上のノードが持つ参照ベクトルと抽出した特徴ベクトルとのユークリッド距離を算出する．算出した距離が最小のノードを勝者ノードとして選択し，勝者ノードとその周辺のノードの参照ベクトルを抽出した特徴ベクトルで更新することにより，逐次学習する．そして，勝者ノードが異常データを学習していた場合に異常を検出する．

文献 [10] では，パケットのヘッダから抽出した特徴量を用いて決定木と SOM を学習させ，それらを組み合わせることで攻撃の検出と逐次学習を行う手法を提案している．パケットのヘッダから特徴量を抽出し，あらかじめ学習を行った決定木を用いて正常と異常のラベル付けを行う．抽出した特徴量とラベル付けの結果を SOM で逐次学習するとともに異常を検出する．また，SOM による異常検出結果により定期的に決定木を再構築する．この手法では逐次学習に対応した SOM と決定木を定期的に再構築することでサイバー攻撃の変化に対応した持続的な攻撃の検出を実現している．しかし，決定木の更新を逐次実施していないことから決定木が攻撃の変化に対応できず精度が低下する可能性が考えられる．

3.　提案手法

本稿では，パケットのヘッダから抽出した特徴量のエントロピーを求めて特徴ベクトルを生成し，文献 [5] で紹介されているエントロピーベース多次元マハラノビス距離に基づく外れ値検出モデルと SOM により特徴ベクトルを逐次学習しながら異常を検出する手法を提案する．本手法では，外れ値検出モデルと SOM の両方で逐次学習を行うことにより，攻撃の変化に対応して精度を維持したまま攻撃を検出できると考えられる．

本手法の異常検出手順について述べる．手法の流れを図 1 に示す．本手法は 2 つの学習フェーズと検出フェーズに分かれている．学習フェーズ 1 では正常なトラフィックデータから抽出した特徴ベクトルを用いて外れ値検出モデルを学習する．学習フェーズ 2 では，正常なデータと攻撃を含むデータのそれぞれに手作業で正常と異常のラベル付けをしておき，抽出した特徴ベクトルを SOM で学習する．また，SOM 学習時に各ノードで，正常の特徴ベクトルを学習した回数と異常の特徴ベクトルを学習した回数をそれぞれ数えておく．

検出フェーズではまず，学習時とは異なる期間に収集したトラフィックデータから抽出した特徴ベクトルに，外れ値検出モデルを適用して正常であるか外れ値であるかの検出を行う．次に，SOM により抽出した特徴ベクトルの逐次学習と異常検出を行う．SOM で選択された勝者ノードが，正常の特徴ベクトルでの学習回数が多い場合，抽出した特徴ベクトルは正常であると認識し，異常の特徴ベクトルでの学習回数が多い場合，異常であると認識する．また，認識した正常/異常の情報を基に外れ値検出モデルを逐次更新する．

3.1　学習フェーズ 1

学習フェーズ 1 では，正常なトラフィックデータから特徴を抽出して学習する．まず注目しているネットワークに対する攻撃を検出するために，ネットワークからトラフィックデータを収集する．収集したトラフィックデータから一定のパケット数 n ごとに表 1 に示すパケットのバイ

表 1　ヘッダから抽出した特徴量
Table 1　Packet Header Features.

送信元 IP アドレス
送信先 IP アドレス
送信元ポート番号
送信先ポート番号
パケットバイト数
プロトコル
TTL 値
フラグメント ID
フラグメント用フラグの状態

ト数などの 9 種類の特徴量を抽出する．区間 i における j 番目の特徴量が H 種類出現した時，h 番目の特徴量の出現回数 $x_i^{j,h}$ から出現確率 $P_i^{j,h} = x_i^{j,h}/n$ を算出し，エントロピー $E_i^{j,h}$ を次式で算出する．

$$E_i^{j,h} = -P_i^{j,h} \log_2 P_i^{j,h} (1 \leq j \leq 9) \tag{1}$$

その後，次式でエントロピーの総和 E_i^j を算出する．

$$E_i^j = \sum_h^H E_i^{j,h} \tag{2}$$

収集した全てのトラフィックデータの区間からエントロピーを算出すると，抽出したエントロピーの値は，特徴量ごとに大きく異なると考えられるため，各特徴量の値の平均が 0，分散が 1 になるように次式で標準化する．

$$E_i^{'j} = \frac{E_i^j - \overline{E^j}}{\sigma_{E^j}} \tag{3}$$

ここで，$E_i^{'j}$ は標準化後のエントロピーを示し，$\overline{E^j}$ と σ_{E^j} は j 番目のエントロピーの平均と標準偏差を表す．そして，標準化したエントロピーで表すベクトル $\boldsymbol{\omega}_i = (E_i^{'1}, E_i^{'2}, \cdots, E_i^{'9})$ を特徴ベクトルとする．

外れ値検出モデルは，学習フェーズ 1 で使用している全てのトラフィックデータから抽出した特徴ベクトルの平均ベクトル $\boldsymbol{\mu}$ と分散共分散行列 $\boldsymbol{\Sigma}$ で表す．

3.2　学習フェーズ 2

学習フェーズ 2 ではまず，正常と異常を含むトラフィックデータから学習フェーズ 1 と同様に特徴ベクトル $\boldsymbol{\omega}_i'$ を抽出し，抽出した特徴ベクトルのそれぞれに手作業で正常と異常のラベルを付与しておく．

次に抽出した特徴ベクトル $\boldsymbol{\omega}_i'$ を SOM で学習する．SOM は，コホネンによって提唱された教師なしニューラルネットワークである [11]．SOM の概要を図 2 に示す．SOM は入力された高次元データが性質の類似するデータと近くなるように，入力された高次元のベクトル空間を低次元のマップ空間上に写像する．マップ空間は人間が見やすいこ

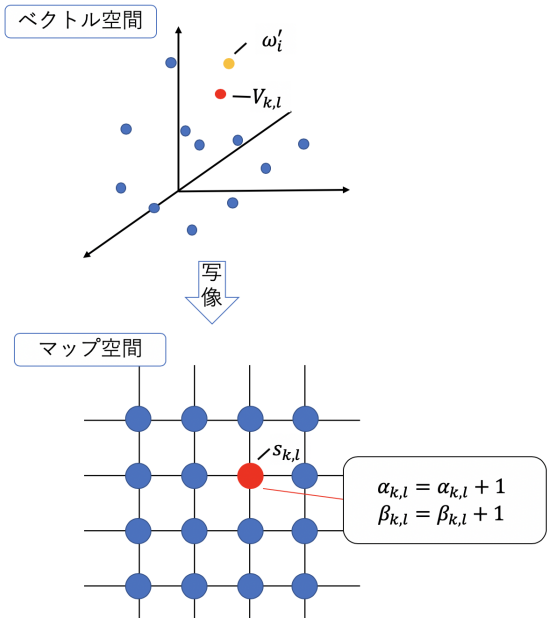

図 2　SOM の概要
Fig. 2　Overview of SOM.

とから 2 次元で用いられることが多く，本稿では 2 次元の SOM を用いることとする．マップ空間上の各ノードは入力データと同次元の参照ベクトルを持ち，参照ベクトルと入力されたデータとのユークリッド距離を用いることで類似度を算出する．

まず，マップ空間上にノード $s_{k,l}$（k,l は任意の整数）を並べる．SOM の精度向上のために文献 [10] では SOM の初期化を行っている．本稿でも SOM の初期化を行う．初期化はマップ空間を 2 つに分けて片側は正常ラベルが付与された特徴ベクトル，もう一方は異常ラベルが付与された特徴ベクトルを入力する．SOM の各ノードは参照ベクトル $\boldsymbol{V}_{k,l}$ を持っている．SOM の学習では，抽出した特徴ベクトルと参照ベクトルとのユークリッド距離を算出する．算出したユークリッド距離の中での最小値をとる参照ベクトル $\boldsymbol{V}_{k,l}$ を持つノード $s_{k,l}$ を勝者ノードとする．勝者ノードとその周囲にあるノードの参照ベクトルを式 (4)〜(7) を用いて更新する．

$$\boldsymbol{V}_{k,l} = \boldsymbol{V}_{k,l} + \theta_s * L * (\boldsymbol{\omega}_i' - \boldsymbol{V}_{k,l}) \tag{4}$$

$$L = L_0 * \exp(\frac{t}{\lambda}) \tag{5}$$

$$\theta_s = \exp(\frac{d^2}{2\sigma_s^2}) \tag{6}$$

$$\sigma_s = \sigma_{s0} * \exp(-\frac{t}{\lambda}) \tag{7}$$

ここで，$\boldsymbol{\omega}_i'$ は入力の特徴ベクトル，t は学習する特徴ベクトルの数，λ は学習した特徴ベクトルの総数，d は算出したユークリッド距離，L_0，σ_{s0} は学習 1 回あたりの参照ベ

クトルの更新率や更新する近傍領域の大きさを決定するパラメータである．そして，各ノードに正常の特徴ベクトルを学習した回数 $\alpha_{k,l}$ と，異常の特徴ベクトルを学習した回数 $\beta_{k,l}$ を持たせる．学習した特徴ベクトルに正常ラベルが付与されている場合は $\alpha_{k,l}$ の数を 1 増やし，異常ラベルが付与されている場合は $\beta_{k,l}$ を 1 増やす．以上の処理を全ての特徴ベクトルに対して行い SOM を学習させる．

3.3 検出フェーズ

新たに観測されたトラフィックデータを一定のパケット数 n で分割し，それぞれの区間で 9 次元の特徴ベクトル $\boldsymbol{\omega}_i''$ を抽出する．まず，抽出した特徴ベクトル $\boldsymbol{\omega}_i''$ に学習フェーズ 1 で学習したマハラノビス距離による外れ値検出モデルを用いてラベル付けを行う．図 3 に 2 次元におけるマハラノビス距離を用いた外れ値検出の概要を示す．図 3 に示した楕円内を正常と定義し，楕円の外の場合，すなわち算出したマハラノビス距離が閾値以上の場合に外れ値と判断する．マハラノビス距離は多次元のデータが持つ相関を考慮して距離を測定する手法の 1 つとして定義される．抽出した特徴ベクトル $\boldsymbol{\omega}_i''$ のマハラノビス距離を学習フェーズ 1 で算出した平均ベクトル $\boldsymbol{\mu}$ と分散共分散行列 $\boldsymbol{\Sigma}$ により算出する．特徴ベクトル $\boldsymbol{\omega}_i''$ のマハラノビス距離 m_i は式 (8) で表される．

$$m_i = \sqrt{(\boldsymbol{\omega}_i'' - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\omega}_i'' - \boldsymbol{\mu})} \qquad (8)$$

マハラノビス距離が閾値 θ_m 以上の場合には外れ値，閾値 θ_m 未満の場合には正常としてラベル付けをする．

次に，SOM で特徴ベクトル $\boldsymbol{\omega}_i''$ の逐次学習と異常検出を行う．SOM では，学習した参照ベクトルの中から特徴ベクトルに最も近い参照ベクトル $\boldsymbol{V}_{k,l}$ を持つノードを選び，勝者ノード $s_{k,l}$ とする．勝者ノードとその周辺にあるノードの参照ベクトルを 3.2 節と同様に更新して逐次学習する．そして，勝者ノード $s_{k,l}$ の $\alpha_{k,l}$ と $\beta_{k,l}$ を比べて $\alpha_{k,l}$ の方が大きければ正常，$\beta_{k,l}$ の方が大きければ異常を検出結果とする．図 4 に SOM による異常検出の概要を示している．

その後，外れ値検出モデルによるラベルを基に SOM の $\alpha_{k,l}$ と $\beta_{k,l}$ を更新する．外れ値検出モデルによるラベル

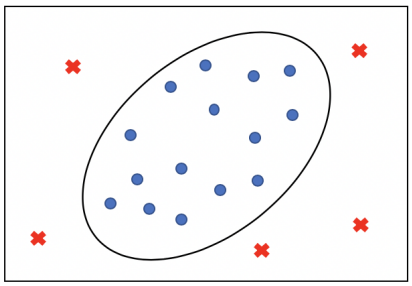

図 3 マハラノビス距離を用いた外れ値検出
Fig. 3 Outlier Detection using Mahalanobis Distance.

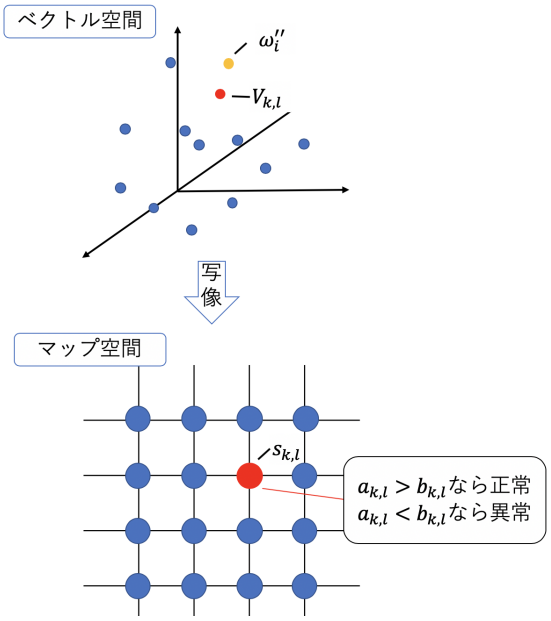

図 4 異常検出の例
Fig. 4 Example of Anomaly Detection.

が正常の場合，$\alpha_{k,l}$ に 1 を追加し，外れ値の場合 $\beta_{k,l}$ に 1 を追加する．さらに，SOM で正常と検出した特徴ベクトルを用いて，外れ値検出モデルの平均ベクトル $\boldsymbol{\mu}$ と分散共分散行列 $\boldsymbol{\Sigma}$ を更新することで，外れ値検出モデルについても逐次学習を行う．

4. 実験

本手法の有効性を確認するために MWS2018 データセットを用いて実験を行った．評価方法は TPR(True Positive Rate) と FPR(False Positive Rate)，F-measure(F 値) を用いた．TPR は異常データを正しく異常とした割合を示し，FPR は正常データを誤って異常と検出した割合を示す．また，F 値は Precision と Recall の調和平均を用いて算出することができる．Precision は異常とした中で実際に異常データであった割合を示し，Recall は異常データを正しく異常とした割合を示す．TPR と FPR，Precision，Recall，F-measure は式 (9)〜(13) で求められる．

$$TPR = \frac{TP}{TP + FN} \qquad (9)$$

$$FPR = \frac{FP}{FP + TN} \qquad (10)$$

$$Precision = \frac{TP}{TP + FP} \qquad (11)$$

$$Recall = \frac{TP}{TP + FN} \qquad (12)$$

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \qquad (13)$$

ここで，TP(True Positive) は異常データを正しく異常とした数，FN(False Negative) は異常データを誤って正常とした数，FP(False Positive) は正常データを誤って異常とした数，TN(True Negative) は正常データを正しく正常とした数を表す．

4.1 実験データ

学習フェーズ 1，学習フェーズ 2，検出フェーズで用いるデータには文献 [12] で紹介されている MWS2018 データセット中の BOS データセットを用いた．BOS データセットは，組織内ネットワークへの侵害活動を想定した動的活動観測のデータセットであり，進行度として 1 から 8 まで定義されている．進行度 1，2 のデータはマルウェアを実行したが通信は発生しておらず，進行度 3，4，5 のデータは通信は発生したが C&C サーバとの通信は成立していない状態を示している．進行度 6，7，8 のデータは通信が発生し，C&C サーバとの通信も成立している状態である．学習フェーズ 1 では，2017 年 8 月 18 日に観測された進行度 2 のトラフィックデータを正常通信データとして使用した．学習フェーズ 2 では，SOM の初期化に 2017 年 8 月 17 日に観測された進行度 2 のトラフィックデータと 2018 年 1 月 23 日に観測された進行度 8 のトラフィックデータをそれぞれ正常通信データと攻撃通信データとして使用した．その後，2018 年 8 月 18 日に観測された進行度 2 のトラフィックデータと 2018 年 1 月 27 日に観測された進行度 8 のトラフィックデータをそれぞれ正常通信データと攻撃通信データとして利用して SOM を学習させた．ここで，学習フェーズ 2 の攻撃通信データは進行度 8 のトラフィックデータから C&C サーバとの通信を抽出したものを用いた．検出フェーズには，2018 年 1 月 28 日に観測された進行度 8 のトラフィックデータをテストデータとして使用した．このトラフィックデータの C&C サーバとの通信を攻撃，それ以外の通信を正常と定義した．

4.2 実験結果

SOM のマップサイズは 40×40 とし，SOM のパラメータ L_0 と σ_{s0} はそれぞれ 0.0001，20 とした，1 区間のパケット数 n は 100 パケットとした．学習フェーズ 2 の学習データの正常通信と攻撃通信の区間数はそれぞれ 6106 区間と 1080 区間だった．また，テストデータの正常通信と攻撃通信の区間数はそれぞれ 5280 区間と 1064 区間だった．正常通信と攻撃通信の区間数が大きく異なるため，異常検出を行う際に誤って正常通信とみなされることが増加すると考えられる．そこで，$\alpha_{k,l}$ と $\beta_{k,l}$ のそれぞれの平均が 0，分散が 1 になるように標準化してから $\alpha_{k,l}$ と $\beta_{k,l}$ を比較することとした．外れ値検出モデルの閾値 θ_m は 7 とした．

実験の結果，提案手法の F 値，TPR，FPR はそれぞれ 0.965，0.961，0.006 となった．

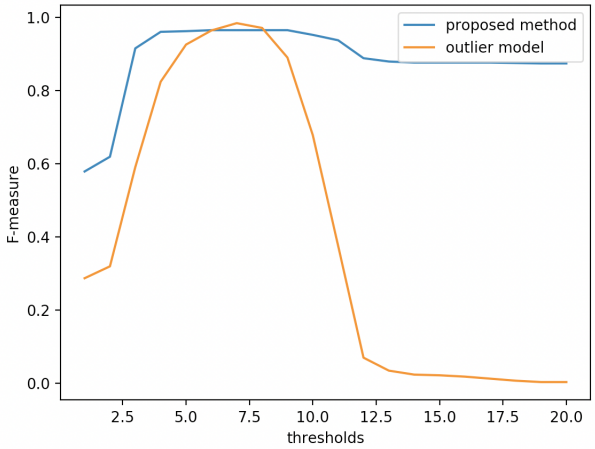

図 5 θ_m の値ごとの F 値
Fig. 5 F-measure when θ_m is changed.

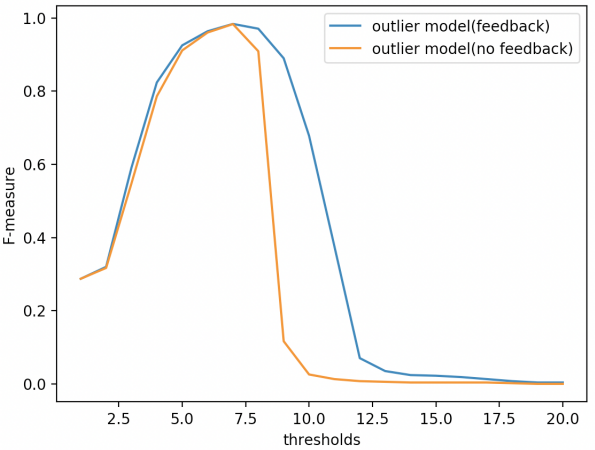

図 6 外れ値検出モデルにおける更新機能を用いた際の F 値の比較
Fig. 6 Comparison of F-measure with and without feedback at outlier model.

4.2.1 実験 1

外れ値検出モデルのラベル付け精度が提案手法に与える影響を確認する実験を行った．外れ値検出モデルの閾値 θ_m の値を 1 から 20 まで変化させた時の提案手法と外れ値検出モデルの F 値の変化を図 5 に示す．F 値は提案手法と外れ値検出モデルの両方で θ_m=7 の時に最も高くなった．θ_m が 1 から 4 までは，外れ値検出モデルの F 値が大きくなると提案手法の F 値も大きくなることから外れ値検出モデルの結果が提案手法の精度に影響していることがわかる．また，θ_m が 9 以上では外れ値検出モデルの精度が低下しているが，提案手法の F 値は外れ値検出モデルよりも下がらなかった．外れ値検出モデルで θ_m が 9 以上の時に F 値が悪化した原因としては，θ_m を大きく設定すると学習した正常通信と特徴が類似しない通信も正常通信とみなすことが多くなったからであると考えられる．提案手法では θ_m が 4〜12 の間で高い F 値を示している．F 値が下がらなかった要因としては，SOM の異常検出結果を基に外

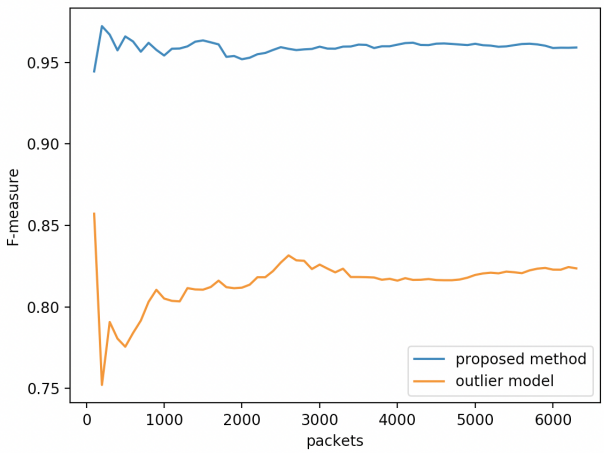

図 7　提案手法と外れ値検出モデルの比較 ($\theta_m = 4$)
Fig. 7　Conparison between proposed method and outlier detection model($\theta_m = 4$).

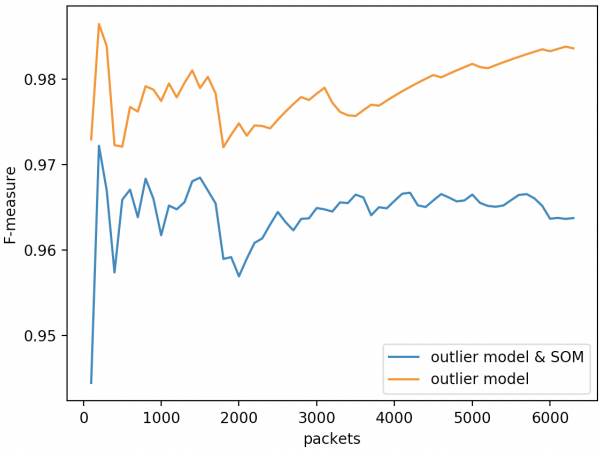

図 8　提案手法と外れ値検出モデルの比較 ($\theta_m = 7$)
Fig. 8　Conparison between proposed method and outlier detection model($\theta_m = 7$).

れ値検出モデルの $\boldsymbol{\mu}$ と $\boldsymbol{\Sigma}$ を逐次更新しているため，θ_m の影響を過度に受けることなく正常な特徴ベクトルの特徴を学習できたためであると考えられる．

次に，SOM の検出結果を用いて外れ値検出モデルを逐次学習させる (更新機能) ことによる外れ値検出モデルのラベル付け精度を確認する実験を行った．更新機能を用いない場合，外れ値検出モデルは外れ値検出モデル自身のラベル付け結果を用いて逐次学習を行う．更新機能を用いた場合と用いなかった場合の外れ値検出モデルの F 値の変化を比較した結果を図 6 に示す．更新機能を用いない場合，$\theta_m = 9$ から F 値が低下している．一方，更新機能を用いた場合，$\theta_m = 10$ から F 値が低下している．更新機能を用いた方が F 値の減少度合いが緩やかになることを図 6 より確認できる．また，更新機能を用いた方が TPR の値の平均が 0.085 高くなり，FPR の値の平均が 0.005 低くなった．TPR の値に差が出たのは，SOM が異常な通信の特徴

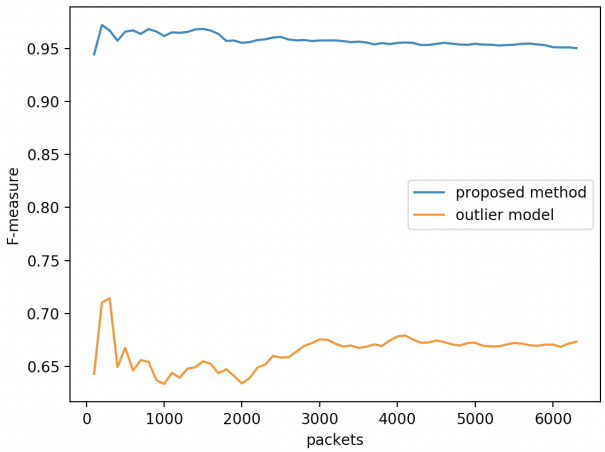

図 9　提案手法と外れ値検出モデルの比較 ($\theta_m = 10$)
Fig. 9　Conparison between proposed method and outlier detection model($\theta_m = 10$).

表 2　検出結果
Table 2　Result of Detection.

		F-measure	TPR	FPR
θ_m=4	提案手法	0.959	0.960	0.008
	外れ値検出モデル	0.824	1	0.086
θ_m=7	提案手法	0.957	0.947	0.007
	外れ値検出モデル	0.974	0.999	0.011
θ_m=10	提案手法	0.950	0.935	0.006
	外れ値検出モデル	0.673	0.521	0.002

をうまく捉えて異常を検出することができ，その検出結果を用いて外れ値検出モデルを逐次学習することができたからだと考えられる．

4.2.2　実験 2

提案手法と外れ値検出モデルのみでの検出結果の比較を行う．閾値 θ_m を 4，7，10 として区間数を 100 ずつ増加させ，提案手法における検出結果と外れ値検出モデルのみを用いた場合の F 値の平均の推移を図 7，8，9 に示す．θ_m=4，10 の時は提案手法の方が精度が高く，θ_m=7 の時は提案手法の方が精度が低いことを確認できた．また，θ_m=4，7，10 の時の提案手法と外れ値検出モデルにおける F 値，TPR，FPR の平均を表 2 に示す．θ_m=4 の時には提案手法の FPR の値が外れ値検出モデルより低く，正常データを正しく検出できており，θ_m=10 の時には提案手法の TPR の値が外れ値検出モデルより高く異常データを正しく検出できている．そのため提案手法での F 値が大きくなったと考えられる．今回のデータでは，θ_m=7 の時に外れ値検出モデルの方が精度が良かったがそれ以外の θ_m では提案手法の方が良かった．実環境で異常検出する時には θ_m の変化にロバストな提案手法の方が良いと考えられる．今後，別のデータを使った実験などによって本手法の

更なる有効性を確認したい．

5. まとめ

本稿では，パケットのヘッダから抽出した特徴量からエントロピーを算出し，生成した特徴ベクトルに対して逐次学習を行う外れ値検出モデルと SOM を適用し，それらを組み合わせることで攻撃の変化に対応して持続的に異常を検出できる手法を提案した．実験では，MWS データセットを用いて本手法の有効性を確認した．実験の結果，逐次学習を行うことでラベル付けの精度が向上することを確認した．さらに，外れ値検出モデルと提案手法を比較した結果，提案手法の方がロバストに攻撃通信を検出可能であることを確認できた．しかし閾値を最適化した場合には提案手法の方が低い結果となる場合があった．今後の課題としては，SOM のパラメータの調整，抽出する特徴量の検討などが挙げられる．また今後，逐次学習を行うことで攻撃の傾向が変化する状況でも精度を維持できることの確認を行う実験を行う予定である．

参考文献

[1] Snort, <https://www.snort.org/>(参照 2018-08-09).

[2] Suricata, <http://suricata-ids.org/>(参照 2018-08-09).

[3] The Bro, <http://www.bro.org/>(参照 2018-08-09).

[4] 山田明, 三宅優, 田中俊昭：亜種攻撃を検知できる侵入検知システム，電子情報通信学会技術報告，ISEC2004-31，pp.119-126(2004).

[5] 小島俊輔, 中嶋卓雄, 末吉敏則：エントロピーベースのマハラノビス距離による高速な異常検知手法，情報処理学会論文誌, Vol.52, No.2, pp.656-668(2011).

[6] 平松尚利, 和泉勇治, 角田裕：複数の通常状態を用いたネットワーク異常検出, 電子情報通信学会技術報告, CS2006-32, pp.61-66(2006).

[7] 佐藤陽平, 和泉勇治, 根元義章：複数の検出モジュールの組み合わせによるネットワーク異常検出の高精度化, 電子情報通信学会技術報告, NS2004-144, pp.45-48(2004).

[8] Mitsuhiro Hatada, and Tatsuya Mori:Finding New Varieties of Malware with the Classification of Network Behavior, IEICE TRANSACTIONS on Information and Systems, vol.E100.D, No.8, pp.1691-1702(2017).

[9] 柿本圭介, 田中英彦：自己組織化マップを用いた Windows システムサービスコールの分類によるマルウェア検出手法, 情報処理学会研究報告, vol.2008, No.45(2008-CSEC-041), pp.43-48(2008).

[10] 小久保博崇, 金岡晃, 満保雅浩, 岡本栄司：攻撃通信検知のための合成型機械学習手法の一検討, 情報処理学会論文誌, Vol.53, No.9, pp.2086-2093(2012).

[11] Kohonen, T.,The self-organizing map, Proceedings of the IEEE, Vol.78, No.9, pp.1464-1480(1990).

[12] 高田雄太, 寺田真敏, 松木隆宏, 笠間貴弘, 荒木粧子, 畑田充弘：マルウェア対策のための研究用データセット MWS 2018 Datasets , 情報処理学会研究報告, Vol.2018-CSEC-82, No.38, pp.1-8(2018).