ネットワークトラフィックからの Stacked Auto-Encoders による低次元特徴抽出と異常検出

Low Dimensional Feature Extraction Using Stacked Auto-Encoders and Anomaly Detection of Network Traffic

日置 裕士 † 谷澤 俊樹 † 青木 茂樹 † 宮本 貴朗 †
Yuji Hioki Toshiki Tanizawa Shigeki Aoki Takao Miyamoto

1. はじめに

近年，サイバー攻撃などのネットワーク犯罪の増加に伴い，ネットワーク上の不正なトラフィックを検出する侵入検知システム (IDS: Intrusion Detection System) の研究が盛んに行なわれている．IDS はシグネチャ型とアノマリ型の 2 種類に大別することができる．代表的なシグネチャ型 IDS として，Snort[1] や Surikata[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．しかしパターンファイルに定義されていない攻撃については亜種を含め検出できないという欠点がある．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，機械学習することで本来検出できない亜種攻撃を検知できる IDS を提案している．しかしこの手法では，学習データをパターンファイルに基づいて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未知の異常については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] 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正常状態と定義し，そこから外れた状態を異常として検出する方式である．しかしこれらのアノマリ型 IDS では異常を検出しても，その異常かどのような異常であるかを判別できないという問題点がある．また正常状態を定義するデータを実運用中のネットワークからは用意できないことも問題点である．

文献 [8] では，パケットのヘッダ情報から抽出した特徴量を用いて構築したアノマリ型 IDS にシグネチャ型 IDS による異常検出結果を組み合わせた手法を提案している．この手法ではアノマリ型 IDS によってクラスタリングし，得られたクラスに対してシグネチャ型 IDS の検出結果をラベルとして付与することで，アノマリ型 IDS では不可能であった攻撃の種類の提示に成功している．しかし対象とするトラフィックによっては全ての攻撃を正しくクラスタリングできない場合があり，クラスタリングの精度が不十分であることによる異常の検出精度の低下が課題となっていた．

一方，文献 [9] では，ネットワークから抽出した特徴量を Stacked Auto-Encoders(SAE) で学習することで，特徴量を低次元に変換する手法を提案している．この手法では変換された特徴量を可視化することで正常な通信か，異常な通信かを分離した状態で 3 次元空間に集約しているが，異常の検出までは行っていない．

そこで本研究では，文献 [8] の手法に文献 [9] で用いられている SAE を組み込むことによってクラスタリング性能を向上させ，攻撃の検出精度を向上させることを目的とする．また DARPA データセットおよび大阪府立大学のインターネットトラフィックを用いて実験を行い，提案手法の有効性を確認した．

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] とアノマリ型 IDS とシグネチャ型 IDS を組み合わせた手法である文献 [8] について述べる．文献 [5] では，パケットのエントロピーに基づく異常検出手法が提案されている．この手法ではまず，IP アドレスやポート番号など毎の単位時間当たりのパケット数を計測する．次に，パケットの発生確率を求め，求めた発生確率からエントロピーを算出する．その後，エントロピーの時系列変化に着目した EMMM 法により，エントロピーが大きく変化する時間を攻撃などが含まれている異常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，複数の正常状態を定義し，各状態との違いから異常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

† 大阪府立大学, Osaka Prefecture University

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 [5,6,7] の手法では，異常が発生したことを検出することはできるものの，発生した異常がどのような攻撃であるかを判断することができないことが問題となっていた．そこで文献 [8] では，パケットのヘッダ情報から抽出した特徴量を用いて構築したアノマリ型 IDS にシグネチャ型 IDS の検出結果を組み合わせることでアノマリ型 IDS だけではできない異常の判別が可能であり，シグネチャ型 IDS だけでは検出できない未知の異常検出が可能な IDS を提案している．この手法ではパケットのヘッダから特徴量を抽出し，主成分分析によって特徴量の次元を圧縮した後，クラスタリングすることでアノマリ型 IDS を構築している．その後，生成された各クラスに対して，シグネチャ型 IDS の検出結果をラベルとして付与することで，従来の手法ではできなかった異常の識別を可能としている．

一方文献 [9] では，ネットワークから抽出した特徴量を Stacked Auto-Encoders(SAE) で学習することで，特徴量を低次元に変換する手法を提案している．この手法では変換された特徴量を 3 次元空間上にプロットすることにより，正常な通信と異常な通信が分離されている状態を可視化することに成功している．また SAE の活性化関数やパラメータを変化させた場合の可視化結果を比較することで，通信を識別できる最適なパラメータを見つけている．しかし，この文献では特徴量の可視化は行っているものの異常の検出は行っていない．

そこで本研究では，文献 [8] の手法の主成分分析による次元圧縮処理を文献 [9] の手法で提案されている Stacked Auto-Encoders(SAE) に置き換えることによって，クラスタリング性能の向上と異常検出精度の向上を目指す．また大阪府立大学のキャンパスネットワークとインターネットとを接続するファイアーウォールの外側で収集したトラフィックデータを用いて実験することで実ネットワークにおいても有効であることを確認する．

3. 提案手法

提案手法の概要を図 1 に示す．本手法は学習と異常検出の 2 つのプロセスに分かれている．まず学習処理では，学習データとなるトラフィックデータを単位時間で分割して，それを区間とする．分割後，区間から 61 次元の特徴ベクトルを抽出する．次に，61 次元の特徴ベクトルの次元を SAE により圧縮する．その後，次元を圧縮した特徴ベクトルをクラスタリングし，クラスタリングの結果得られた各クラスの重心に最も近い特徴ベクトルの区間に対して，シグネチャ型 IDS を適用し，攻撃の種類を特定するラベルを付与する．異常検出処理では，クラスタリングに用いたデータとは別のトラフィックデータから特徴ベクトルを抽出して，学習した空間に投影し，最も距離の近いクラスのラベルを出力することで正常と異常を識別する．

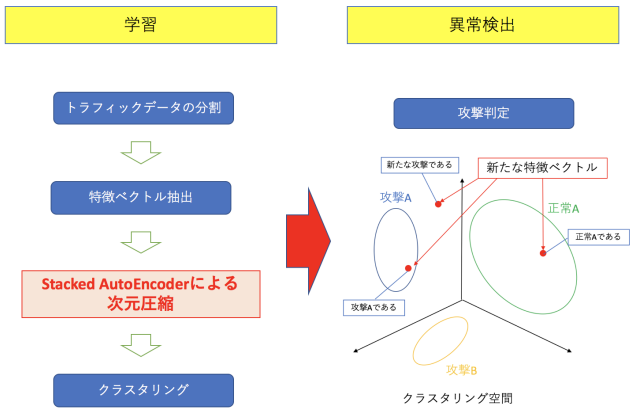

図 1: 提案手法の概要

3.1 パケットからの特徴ベクトルの抽出

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，注目しているネットワークと外部ネットワーク (インターネット) 間の送受信パケットを収集し，単位時間 ω の区間で分割し，N 個の区間でパケットを分割する．収集したパケットのヘッダから，単位時間毎に表 1 に示す 61 種類の特徴量を抽出する．

3.2 Stacked Auto-Encoders(SAE) による次元圧縮

抽出した各区間の特徴ベクトルをクラスタリングする際に，特徴ベクトルの次元が 61 次元と大きいため，次元を圧縮する．本手法では，次元圧縮手法にディープラーニングの学習アルゴリズムの一つである SAE を利用する．SAE は Auto-Encoder(AE) を複数積み上げることで構成され，入力したデータから情報量を削減した別の特徴を獲得できるという性質を持つ．そのため，圧縮前の特徴を維持したまま，特徴ベクトルを低次元に圧縮できる．

3.2.1 Auto-Encoder(AE)

AE とはニューラルネットワークにおける教師なし学習手法のひとつである．他のニューラルネットワーク手法

表 1: 特徴量の一覧

パケットサイズ平均	パケットサイズ最大値
パケットサイズ最小値	パケット到着間隔平均
パケット到着間隔最小時間	パケット到着間隔最大時間
パケット到着間隔分散	パケット到着間隔の総時間
パケットサイズの総数	パケット数
パケットサイズの分散	TTL 値平均
TTL 値分散	宛先 IP アドレス種類数
送信元 IP アドレス種類数	送信元ポート番号種類数
宛先ポート番号種類数	SYN パケット数
FIN パケット数	PSH パケット数
RST パケット数	URG パケット数
ACK パケット数	FIN&ACK パケット数
RST&ACK パケット数	SYN&ACK パケット数
PSH&ACK パケット数	TCP 中の RST 割合
TCP 中の SYN 割合	TCP 中の PSH 割合
TCP 中の URG 割合	TCP 中の FIN 割合
TCP 中の ACK 割合	TCP 中の RST&ACK 割合
TCP 中の PSH&ACK 割合	TCP 中の SYN&ACK 割合
TCP 中の FIN&ACK 割合	ICMP パケット数
UDP パケット数	送信元ポート番号 110 番パケット数
送信元ポート番号 22 番パケット数	送信元ポート番号 53 番パケット数
送信元ポート番号 443 番パケット数	送信元ポート番号 80 番パケット数
送信元ポート番号 25 番パケット数	送信元ポート番号 465 番パケット数
送信元ポート番号 587 番パケット数	送信元ポート番号 995 番パケット数
送信元ポート番号 993 番パケット数	送信元ポート番号 143 番パケット数
宛先ポート番号 110 番パケット数	宛先ポート番号 22 番パケット数
宛先ポート番号 53 番パケット数	宛先ポート番号 443 番パケット数
宛先ポート番号 80 番パケット数	宛先ポート番号 25 番パケット数
宛先ポート番号 465 番パケット数	宛先ポート番号 587 番パケット数
宛先ポート番号 995 番パケット数	宛先ポート番号 993 番パケット数
宛先ポート番号 143 番パケット数	

と同様に誤差逆伝搬法によって学習を行うが，教師データを持たないため，入力データ自体を教師データとみなし出力層の値を入力データに近づけていくように学習する．このようにして最適化された AE の中間層の値は入力データをよく表した特徴になっているという性質を持つ．この性質を利用して，中間層の次元数を入力層の次元数より小さくすることで入力データを低次元に圧縮する．

具体的な手順を説明する．一般的に AE は入力層，中間層，出力層の 3 層で構成される．入力データのベクトル $\boldsymbol{x}$ をエンコードし，中間層のデータベクトル $\boldsymbol{x'}$ を生成し，さらにそれをデコードすることで出力層のデータベクトル $\boldsymbol{x'}$ を生成する．エンコード，デコードに用いるパラメータをそれぞれ重み $\boldsymbol{W_1}$，$\boldsymbol{W_2}$，バイアス $\boldsymbol{b_1}$，$\boldsymbol{b_2}$ とする．これらより以下のように表される．

$$\boldsymbol{h} = \boldsymbol{W_1}\boldsymbol{x} + \boldsymbol{b_1} \tag{1}$$

$$\boldsymbol{x'} = \boldsymbol{W_2}\boldsymbol{h} + \boldsymbol{b_2} \tag{2}$$

重みの初期値はランダムに生成した値を用いることが一般的であるため，$\boldsymbol{W_1}, \boldsymbol{W_2}$ は正規分布の乱数によって初期値を与え，$\boldsymbol{b_1}, \boldsymbol{b_2}$ は 0 で初期化する．また活性化関数として Softplus 関数を利用する．

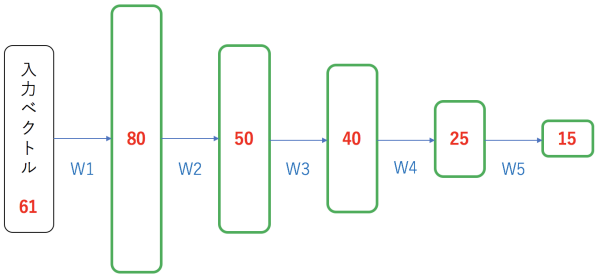

図 2: 本手法の SAE の概要

入力データを再現するためにこれらのパラメータを最適化していくことが AE の学習の目的であるため，以下に表される誤差関数 E を最小化していくことで学習を行う．

$$E(x, x') = -\sum x'_i \log x_i \tag{3}$$

誤差関数 E を最小化するために確率的降下勾配法 (SGD) を用いてミニバッチ学習を行うことでパラメータを最適化する．

3.2.2 Stacked Auto-Encoder(SAE)

学習した AE の中間層のデータベクトルを入力ベクトルとみなし，さらに AE で学習する．この手順を複数回繰り返していくことで AE を積み重ねたものを SAE と呼ぶ．本手法の SAE の概要を図 2 に示す．また文献 [9] の手法では，各層の AE の学習の際には Softplus 関数を用いたが，学習後，次の層の入力値となる出力値を計算する場合には Sigmoid 関数を用いた場合の方が良い結果を得られたとされているため，本手法においても同様とする．このようにして最終的に得られたデータベクトルは入力データの必要な情報のみを維持した低次元の特徴ベクトルに圧縮されている．

3.3 特徴ベクトルのクラスタリング

特徴ベクトル $\boldsymbol{I}_t$ と $\boldsymbol{I}_{t+1}$ の 2 つの区間において同様の攻撃が含まれるとき，特徴量が類似するために特徴ベクトル間の距離は小さくなる．一方，性質の異なる攻撃を含む特徴ベクトル間では距離が大きくなる．そのため，抽出した特徴ベクトルをクラスタリングすると，同一クラスに分類された特徴ベクトル同士では，類似した攻撃を含むと考えられる．本手法では SAE によって次元を圧縮した特徴ベクトルを Mean-Shift 法によりクラスタリングする．この手法はあらかじめ分類するクラス数を定めないため，今回のような異常の種類数が判明していない場合等に適している．

3.4 各クラスへのラベル付け

クラスタリングされた各クラスにどのような攻撃が含まれているかをシグネチャ型 IDS を用いて調べ，クラスのラベルとして利用する．各クラスの重心に最も近い特徴ベクトルを選択し，その区間に対してシグネチャ型 IDS を用いて異常検出を行う．出力された結果をそのクラスのラベルとして付与することで，そのクラスがどのような攻撃を含むかを表す．ここでシグネチャ型 IDS のパターンファイル中の v 番目のルールで $s'_{t,v}$ 回の異常が検出された場合，$(s'_{t,1}, s'_{t,2}, \cdots, s'_{t,v}, \cdots, s'_{t,q})$ をラベルとして付与する．ここで，q はシグネチャ型 IDS のパターンファイルで定義されているルールの総数である．このラベル付与を全てのクラスに行うことで異常検出および攻撃の識別が可能な IDS を構成することができる．ここで正常な通信のみを含むクラスについては，全てラベルが $\mathbf{0}$ ベクトルとなるが，別の状態を表すクラスとして扱うことにしている．

3.5 異常検出

新たに観測されたトラフィックデータを単位時間 ω で分割しそれぞれの区間から 61 次元の特徴ベクトルを抽出する．学習時と同様に SAE によって 61 次元の特徴ベクトルを低次元に圧縮する．圧縮された特徴ベクトルと 3.3 節のクラスタリングによって得られた各クラスの重心との距離を f とし，最も距離が近いクラスを選択する．f がしきい値未満の場合には，そのクラスに属すると判断し，そのクラスのラベルを検出結果として出力する．しきい値以上の場合にはそのクラスに属さないと判断し，新たな異常として識別する．

4. 実験

本手法により高精度なクラスタリングと異常検出ができることを確認する実験を行った．まず DARPA データセットを用いた予備実験により従来手法 [8] と比較する．次に大阪府立大学のインターネットトラフィックデータを用いてクラスタリングおよび異常検出を行い，実ネットワークにおいても異常を検出できることを確認する．

4.1 予備実験

まず DARPA データセットを用いた予備実験を行った．DARPA データセットには攻撃の種類と攻撃開始時刻の情報が含まれているため，その情報を用いて従来手法 [8] との比較を行う．そのため，3.4 節で述べたシグネチャ型 IDS によるラベル付けは行っていない．学習用としては DARPA1999outside データセットの 1999 年 3 月 9 日午前 8 時 00 分 01 秒から 1999 年 3 月 10 日午前 2 時 59 分 59 秒までのデータを使用した．単位時間は 1 分とした．テストデータとしては 1999 年 3 月 10 日午前 8 時 00 分 03 秒から 1999 年 3 月 11 日午前 6 時 00 分 01 秒までのデータを使用し，学習時と同様に単位時間は 1 分とした．学習データおよびテストデータに含まれる攻撃の種類と区間数を表 2, 表 3 に示す．

表 2: 学習用データ

攻撃名	区間数
正常区間	1046
portsweep	28
ipsweep	26
mailbomb	10
loadmodule	2
eject	2
httptunnel	2
secret	1
phf	1

表 3: テストデータ

攻撃名	区間数
正常区間	1264
ipsweep	14
mailbomb	11
satan	3
perl(Failed)	1
crashiis	1

4.1.1 クラスタリング実験および考察

DARPA データセットを用いて，クラスタリング実験を行った．本手法を用いた場合の攻撃ごとのクラスタリング結果を表 4 に示す．この表より portsweep を含む区間の 28 区間中 26 区間は portsweep のみを含むクラスと分類された．また mailbomb を含む区間はすべての区間が mailbomb のみを含むクラスと分類された．eject を含む区間も同様にすべての区間が eject のみを含むクラスと分類された．また ipsweep を含む区間は 26 区間中 11 区間が loadmodule, phf と同じクラスとして分類され，残りの 15 区間は正常のクラスと分類された．httptunnel, secret に関しては攻撃のクラスと分類することはできなかった．

また本手法によるクラスタリング実験結果と文献 [8] の手法のクラスタリング実験結果の比較を表 5 に示す．この表では本手法と従来手法を用いて実験を行った場合の双方で，クラスタリングの結果得られた全クラス数，正常区間のみのクラス数，異常区間のみのクラス数，正常区間と異常区間が混在しているクラス数を示している．実験の結果，従来手法で正しく攻撃と分類されていた portsweep, ipswep, mailbomb の 3 種類の攻撃に加え，従来手法では攻撃と分類されなかった loadmodule, eject, phf の 3 種類の攻撃を本手法を用いることで攻撃と分類することに成功した．

portsweep を含む区間は，従来手法では 7 個，本手法では 4 個のクラスに分類された．portsweep は攻撃対象となるホストの侵入に利用可能なポートを探すため，多数のポートに連続的にアクセスする攻撃である．そのため，本手法の特徴ベクトルの中でも宛先ポート番号の種類数の特徴量に大きな変化が表れた．これが portsweep を含む区間を特定のクラスに分類できた要因と考えられ

る．

mailbomb を含む区間は従来手法，本手法ともにすべて 1 個のクラスに分類された．mailbomb とは特定のメールサーバに対して大量のメールを送信する攻撃である．SMTP を通じてメールを送信する際，25 番ポートを利用するため，本手法の特徴ベクトルの中でも宛先ポート番号 25 番のパケット数の特徴量に大きな変化が表れた．これが mailbomb を含む区間を特定のクラスに分類できた要因と考えられる．

ipsweep は対象となるネットワークに対して，攻撃対象となるホストを探し出すため，IP アドレスを変化させながら連続的に通信を行う攻撃である．抽出した 61 種類の特徴量を確認すると，宛先 IP アドレスの種類数の特徴量に大きな変化が表れていることがわかった．しかし実験の結果，従来手法ではすべての区間が正しく分類されていたが，本手法では ipsweep を含む区間の内，全 26 区間中の 15 区間が正常区間のクラスに分類された．これは 61 次元の特徴量では変化が確認できたことから，SAE による次元圧縮においてクラスタリングに必要な特徴が欠落してしまったことが原因と考えられる．そのため SAE の構築の際のパラメータを調整していくことが今後の課題として挙げられる．また 61 種類の特徴量を再度検討し，特徴量を取捨選択することも課題として挙げられる．

また従来手法では攻撃のクラスとして分類されなかった攻撃である loadmodule, eject, phf の 3 種類は本手法ではそれぞれ攻撃のクラスとして分類することができた．eject は独立したクラスとして分類されたが，loadmodule, phf は ipsweep の一部と同一のクラスと分類された．これは ipsweep と loadmodule, phf に類似した特徴が表れていたことが原因と考えられる．このように従来手法では正常と分類されていた攻撃を本手法では攻撃と分類できた要因として，従来手法の主成分分析においては，主成分得点が低くなっていた特徴を本手法の SAE では有効な特徴として使用できたためであると考えられる．3 種類の攻撃 (secret, back, httptunnel) は従来手法と同様に正常区間のクラスに誤って分類されたが，これらの攻撃はパケットのヘッダに特徴が表れない攻撃であったため本手法では攻撃のクラスに分類することができなかった．

これらの結果から，本手法ではパケットのヘッダに特徴が表れる攻撃であれば，攻撃を分類することが可能であるとわかった．また従来手法では分類できなかった攻撃をクラスタリングできたことから，本手法の SAE を用いた次元圧縮の有効性を確認することができた．

表 4: 攻撃のクラスに分類された区間数

攻撃名	区間数
portsweep	26/28
ipsweep	11/26
mailbomb	10/10
loadmodule	2/2
eject	2/2
httptunnel	0/2
secret	0/1
phf	1/1

表 5: 従来手法とのクラスタリング結果の比較

	全クラス数	正常区間	異常区間	正常区間と異常検出
従来手法	29 個	16 個	portsweep(7 個) ipsweep(4 個) mailbomb(1 個)	1 個
本手法	24 個	14 個	portsweep(4 個) mailbomb(1 個) ipsweep+loadmodule+phf(1 個) eject(2 個)	3 個

4.1.2 異常検出実験および考察

DARPA データセットを用いて，異常検出実験を行った．テストデータから抽出した特徴ベクトルを前節でクラスタリングした空間に投影し，クラスとの距離を基に異常を検出した．実験の結果を表 6 に示す．この表では検出に成功した攻撃，検出に失敗した攻撃，新たな異常として検出された攻撃を表している．また本手法による異常検出実験結果と文献 [8] の手法の異常検出実験結果の比較を表 7 に示す．

まず学習用データと共通して含まれる攻撃である mailbomb を含む区間はすべての区間において，学習用データの mailbomb を含む区間と同一のクラスであると正しく検出できた．テストデータにもクラスタリングデータと同様の傾向が見られたため，同一のクラスに分類された．ipsweep を含む区間は 14 区間中 12 区間が正常区間と誤って検出された．残りの 2 区間は新たな異常として検出された．従来手法においてもいくつかの区間を新たな異常として検出したが ipsweep と検出することはできなかった．これについては学習データに含まれる ipsweep の特徴とテストデータに含まれる ipsweep の特徴に違いがあったことが原因と考えられる．

次にクラスタリングデータに含まれない未知の攻撃のである satan については従来手法と同様に新たな異常として検出された．perl(Failed) についてはクラスタリング時と同様に，パケットのヘッダに特徴が表れない攻撃であったことが検出失敗となった原因と考えられる．crashiis は従来手法では検出されなかったが本手法では

表 6: 異常検出結果

攻撃名	検出成功	検出失敗	新たな異常
ipsweep	0/14	12/14	2/14
mailbomb	11/11	0/11	0/11
satan	0/3	0/3	3/3
perl(Failed)	0/1	1/1	0/1
crashiis	0/1	0/1	1/1

表 7: 従来手法との異常検出結果の比較

	検出成功	検出失敗	新たな異常
従来手法	mailbomb	perl(Failed) crashiis	satan ipsweep
本手法	mailbomb	perl(Failed)	satan ipsweep crashiis

新たな異常として検出できた．従来手法では使用されていなかった特徴量を本手法の SAE による次元圧縮では有効な特徴量として使用していたため crashiis を新たなクラスとして検出したと考えられる．

これらの結果から，SAE による次元圧縮を行うことで，従来手法では検出できなかった，学習データに含まれない未知の攻撃である crashiis を検出することに成功したことを確認した．しかし従来手法では検出できていた ipsweep を検出できなかったという問題点もあるため，今後より検出精度を高められるように抽出する特徴量を取捨選択する必要があると考えられる．

4.2 実ネットワークにおける実験

大阪府立大学のキャンパスネットワークと，インターネッ トとを接続するファイアーウォールの外側でトラフィックを収集し実験を行った．まず学習用データとして，2016 年 7 月 20 日 13 時 16 分 50 秒から 1 時間のトラフィックを収集し，単位時間を 1 秒として 3600 個の区間に分割し，特徴ベクトルを抽出した．各クラスへのラベル付けにはシグネチャ型 IDS の一つである Snort[3] で 2017 年 7 月 18 日に取得したルールを使用した．次に異常検出用データとして，2016 年 8 月 4 日 16 時 43 分 00 秒から 1 時間トラフィックを収集し，学習用データと同様に単位時間を 1 秒として 3600 個の区間に分割し，特徴ベクトルを抽出した．またトラフィックの収集方法は，tcpdump を用いてパケットをキャプチャし pcap 形式のファイルで保存することにより行った．

4.2.1 クラスタリング実験および考察

4.2 節で述べた学習用データを用いてクラスタリング実験を行った．実験の結果，3600 個の区間は 33 個のクラスに分類された．表 8 はあるクラスにつけられたラベルの一例を示している．ラベルはクラスの重心に最も近い区間に含まれていた攻撃名とその個数をまとめたものである．表 8 よりこのクラスでは 7 種類の攻撃が含まれていることがわかった．また表 8 で示したクラスとは別のクラスにつけられたラベルを表 9 に示す．表 8 のクラスの結果と比べると表 9 のクラスでは表 8 のクラスに含まれる攻撃に加え，さらに 4 種類の攻撃が含まれていた．これによって表 8 のクラスとは別のクラスに分類されたと考えられる．また表 9 のクラスに分類された他の区間も同様に snort を適用するとすべての区間において共通して含まれる攻撃がいくつかあった．その中でも (POP) Unknown POP3 command はデータセット全体において比較的含まれる個数が少ないにもかかわらず，表 9 のクラスに属するすべての区間に含まれていた．よってこのクラスは (POP) Unknown POP3 command を表したクラスではあると考えられる．この結果から実際に収集したデータからも DARPA データセットを用いた場合と同様に攻撃ごとに通信を分類することが可能であるとわかった．

表 8: クラスのラベル例 1

攻撃名	個数
Consecutive TCP small segments exceeding threshold	61
(http inspect) LONG HEADER	86
(http inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	48
(http inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE	11
(http inspect) UNESCAPED SPACE IN HTTP URI	2
Reset outside window	22
TCP Timestamp is missing	12

表 9: クラスのラベル例 2

攻撃名	個数
Consecutive TCP small segments exceeding threshold	67
(http inspect) LONG HEADER	52
TCP Timestamp is outside of PAWS window	50
(http inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	44
(http inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE	29
(http inspect) UNESCAPED SPACE IN HTTP URI	8
Reset outside window	8
TCP Timestamp is missing	4
Limit on number of overlapping TCP packets reached	1
(spp ssl) Invalid Client HELLO after Server HELLO Detected	1
(POP) Unknown POP3 command	1

4.2.2 異常検出実験および考察

次に 4.2 節で述べた異常検出用データを用いて異常検出実験を行った．実験の結果，3583 区間が学習データと同じクラスであると検出され，17 区間は新たな異常として検出された．クラスタリング実験において確認した (POP) Unknown POP3 command のクラスと検出された 11 区間に snort を適応した結果，5 区間から (POP) Unknown POP3 command を含むことが確認できた．また残りの 6 区間から 1 区間を選び，snort を適用した結果を表 10 に示す．この表より (POP) Unknown POP3 command は含まれていないが，それ以外の攻撃が類似

表 10: snort の適用結果の例 1

攻撃名	個数
Consecutive TCP small segments exceeding threshold	85
(http inspect) LONG HEADER	90
TCP Timestamp is outside of PAWS window	14
(http inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	50
(http inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE	17
(http inspect) UNESCAPED SPACE IN HTTP URI	1
Reset outside window	18
TCP Timestamp is missing	4
(spp ssl) Invalid Client HELLO after Server HELLO Detected	1
Bad segment, adjusted size ¡= 0	6

表 11: snort の適用結果の例 2

攻撃名	個数
Consecutive TCP small segments exceeding threshold	86
(http inspect) LONG HEADER	55
TCP Timestamp is outside of PAWS window	10
(http inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	74
(http inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE	9
(http inspect) UNESCAPED SPACE IN HTTP URI	3
Reset outside window	10
TCP Timestamp is missing	12
(spp ssl) Invalid Client HELLO after Server HELLO Detected	1
Bad segment, adjusted size ¡= 0	2
(http inspect) SERVER CONSECUTIVE SMALL CHUNK SIZES	1
(spp frag3) Fragmentation overlap	85

していたため同じクラスとして検出されたと考えられる．また新たな異常として検出された 17 区間の一つに snort を適用した結果を表 11 に示す．表 11 を見ると (spp flag3) Fragmentation overlap が多く含まれていることがわかる．これは学習データにあまり含まれていない攻撃であるため，その影響により新しい異常と検出されたと考えられる．

5. まとめ

本論文では，パケットのヘッダから特徴ベクトルを抽出し，SAE によって次元圧縮した後，クラスタリングを行うことで類似する通信を識別する手法を提案した．DARPA データセット，大阪府立大学において収集したトラフィックデータを用いた SAE の学習，及び異常検出実験を行い，本手法の有効性を確認した．

実験の結果，従来手法では正しく分類できなかった攻撃を分類することができた．また従来手法では検出できなかった未知の異常を検出することにも成功した．今後の課題として，抽出した特徴量の検討や，SAE のパラメータ調整などが挙げられる．

参考文献

[1] Snort https://www.snort.org/

[2] Surikata http://surikata-ids.org/

[3] The Bro http://www.bro.org/

[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, ” 信学技報, ISEC2004-31, 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, ” 情処学論, Vol.52, No.2, 656-668, 2011

[6] 佐藤陽平, 和泉勇治, 根元義章, “複数の検出モジュールによるネットワーク異常検出の高精度化, ” 信学技報, NS2004-144, 2004

[7] 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, ” 信学技報, CS2006-32, 2006

[8] 谷澤 俊樹，青木 茂樹，宮本 貴朗，“ シグネチャ型 IDS とアノマリ型 IDS の組み合わせによる未知の異常検知”, 第 15 回情報技術フォーラム (FIT2016) 講演論文集, 分冊 4, no.L-029, pp.173-174, 2016

[9] 長田元気，西出隆志，”Stacked Denoising Auto-Encoders によるネットワーク攻撃の特徴抽出と可視化”，2016 年暗号とセキュリティシンポジウム (SCIS2016) 講演論文集, 4B2-5, 2016