Computer Security Symposium 2017
23 - 25 October 2017

パケットのヘッダ情報に注目したアノマリ型IDSとシグネチャ型IDSを組み合わせた未知の異常検出

谷澤 俊樹[1]　青木 茂樹[1]　宮本 貴朗[1]

概要：近年，サイバー攻撃への対策として侵入検知システム (IDS) に関する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．シグネチャ型はパターンファイルに定義されていない異常は検出できず，アノマリ型は検出した異常の種類を判別できないという問題がある．そこで本稿では 2 種類の IDS を組み合わせた侵入検知手法を提案する．まずトラフィックデータから特徴量を抽出し，主成分分析法により次元を削減する．次に特徴量をクラスタリングし，各クラスにラベル付けすることで未知の異常の検出及び異常の種類を判別する．実験では，DARPA1999 データセット, MWS データセット, 大阪府立大学ネットワークにおけるトラフィックデータに本手法を適用し，有効性を確認する．

キーワード：MWS, クラスタリング, IDS

Toshiki Tanizawa[1]　Shigeki Aoki[1]　Takao Miyamoto[1]

1. はじめに

近年，インターネットの普及に伴い Web やメール等を通じたサイバー攻撃が増加している．そこでサイバー攻撃への対策として，ファイアーウォール等と共に攻撃を自動で検知するための IDS(侵入検知システム) 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．これらのシグネチャ型 IDS は既に多くの環境において実運用されている． シグネチャ型 IDS はあらかじめ異常を定義した，パターンファイルに基づいて異常の検出を行うために，アノマリ型 IDS と比較すると誤検知が少ないというメリットがある．しかし，パターンファイルに登録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新規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． これらの手法では，正常な通信のみを含むデータを用いて正常状態を定義し， 正常から外れた状態を異常として検出する．しかし，これらのアノマリ型 IDS の手法では異常を検出してもその異常がどのようなものかを判別できない．また，アノマリ型 IDS を構築する際に用いる正常な通信のみを含むデータを，実運用中のネットワークでは用意できないことも欠点として挙げられる．

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ることで，未知の異常検出及び異常の判別が可能な IDS を提案している．しかし，特徴を抽出する際に単位時間を 1 つ設定して分割しているため，単位時間をまたぐ攻撃を正しく検出できない可能性が考えられる．

そこで本稿では，特徴抽出の際に単位時間を複数設定して分割することで 1 つの単位時間では検出が難しい攻撃を識別する．また，Web やメールに関連する特徴を追加し，従来手法では発見できない攻撃の検出を試みる．

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

[1] 大阪府立大学
Osaka Prefecture University

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すべてのクラス 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，異常状態をトラフィッ 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，単位時間当たりのトラフィック量を数値化した特徴量，TCP のフロー毎のフラグの出現回数を数値化した特徴量，TCP の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，異常の種類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，クラスタリングし，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．しかし，トラフィックデータから特徴を抽出する際に単一の単位時間で分割するため，単位時間をまたぐ攻撃では特徴を正しく抽出できず，異常を検出できない可能性がある．そこで本手法ではトラフィックデータから特徴を抽出する際に単位時間を複数設定する

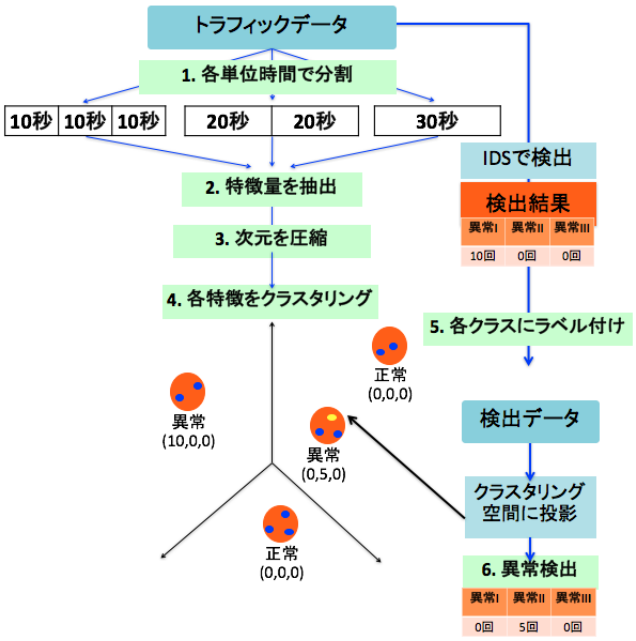

図 1　提案手法の概要

ことで特定の単位時間では検出が難しい攻撃を検出する．また，Web やメールに関連する特徴を追加し，従来手法では発見できない攻撃の検出を試みる．また，DARPA1999 データセット，MWS データセット，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，本手法が有効であることを確認する．

3. 提案手法

本手法の概要を図 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複数の単位時間で分割して，それを区間とする．複数の単位時間で分割した後，各区間から 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を主成分分析法により圧縮する．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同様に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．

3.1 トラフィックデータ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

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	

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，複数の単位時間 $\omega_1, \omega_2, \cdots, \omega_M$($M$ は単位時間の種類数) 毎にトラフィックデータを分割する． 複数の単位時間で分割した各区間から，表 1 に示す 61 種類の特徴量を抽出する．

3.2　クラスタリング

特徴ベクトルを抽出した際に，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異常が含まれるとき，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，性質の異なる異常を含む特徴ベクトル間では距離が大きくなる．そこで，特徴ベクトルをクラスタリングする．まず，61 次元の特徴ベクトルでは次元が大きいため主成分分析法により次元を圧縮する．ここでは，累積寄与率 80%以上となる最小の次元数 r で圧縮する．主成分分析法により圧縮された空間を S 空間と呼ぶと，S 空間上での座標値は，$\boldsymbol{I}_t = (s_{t,1}, s_{t,2}, \cdots, s_{t,r})$ で表される．その後，S 空間上の座標値を基に Mean-Shift 法を用いてクラスタリングする．Mean-Shift 法はあらかじめ分類するクラス数を定めないため，今回のように異常の種類数が判明していない場合に適した手法である．

3.3　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．ラベルづけの概要を図 2 に示す．前節で抽出した特徴ベクトルをクラスタリングした結果，同一クラスに分類された特徴ベクトル同士では，類似した異常を含むと考えられる．そのため，シグネチャ型 IDS でそれらの区間を異常検出した場合においても，類似した結果を出力すると考えられる．そこで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異常を含むかを表す．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． このラベル付与をすべ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すべてのクラスはラベルが $\mathbf{0}$ ベクトルとなるが，別の状態を表すクラスとして扱う．

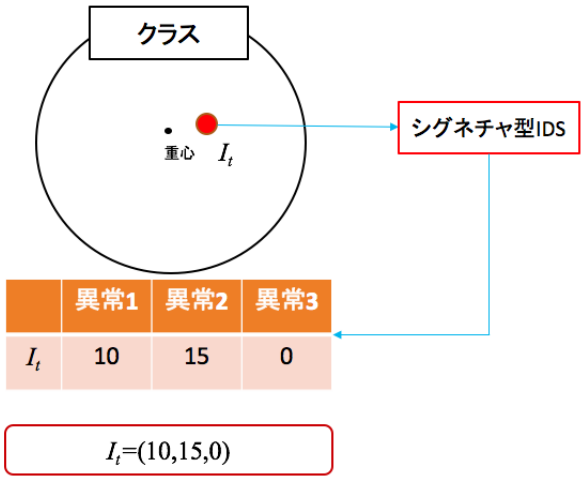

図 2　ラベル付けの概要

3.4　異常検出

異常検出の例を図 3 に示す．異常検出処理では新たに観測されたトラフィックデータを学習データと同様に複数の単位時間 $\omega_1, \omega_2, \cdots, \omega_M$ 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 主成分分析法によって 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 3.2 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，新たな異常と識別する．

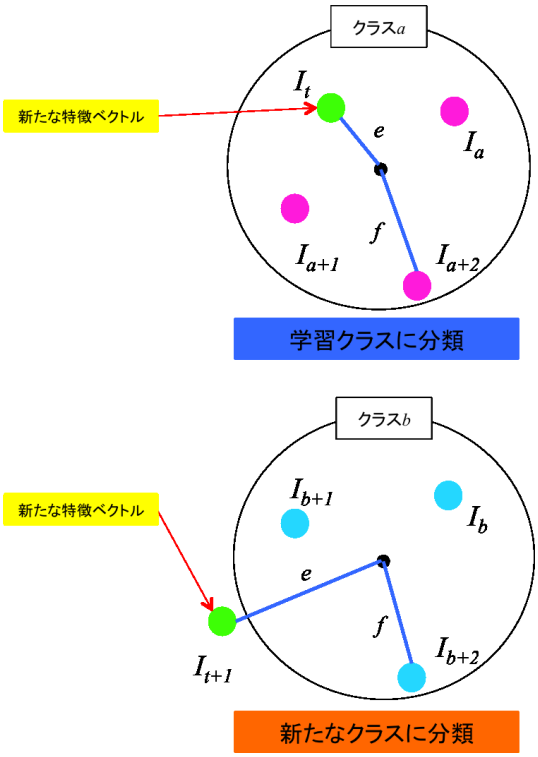

図 3　提案手法の概要

表 2　DARPA データセットに含まれる攻撃名と区間数

攻撃名	区間数 (10 秒)	区間数 (60 秒)
正常通信	6278	1046
portsweep	168	28
ipsweep	156	26
mailbomb	60	10
loadmodule	12	2
eject	12	2
httptunnnel	12	2
secret	6	1
phf	6	1

表 3　DARPA データセットのクラスタリング結果

攻撃名	区間数 (10 秒)	区間数 (60 秒)
portsweep	160/168	26/28
ipsweep	0/156	24/26
mailbomb	0/60	10/10
loadmodule	0/12	0/2
eject	0/12	0/2
httptunnnel	0/12	0/2
phf	0/6	0/1

4.　実験

本手法の有効性を確認するため，DARPA1999 データセットを用いて予備実験を行った．また，MWS2017 データセットおよび大阪府立大学のキャンパスネットワークにおけるトラフィックデータを用いて実験を行った．

4.1　予備実験

DARPA データセットには攻撃の種類と攻撃開始時間の情報が記録されているため，その情報を用いて本手法の有効性を確認する．そのため，3.3 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．クラスタリングに用いるデータセットとして，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．今回は単位時間を 10 秒，60 秒とし，特徴量の抽出およびクラスタリング実験を行った．表 2 に，含まれる攻撃の種類と単位時間ごとの区間数を示す．

4.1.1　実験結果及び考察

クラスタリングの結果を表 3 に示す．10 秒の単位時間では portsweep を特定のクラスに正しく分類できたが，8 個の区間は正常通信と同一のクラスに分類された．一方，ipsweep や mailbomb ではすべての区間が正常通信と同一のクラスに分類された．しかし，単位時間を 60 秒とした区間については，それぞれのクラスに正しく分類できていた．これは，単位時間を 10 秒とした場合では変化が現れなかった宛先 IP アドレス種類数等の特徴量が 60 秒の場合には他の区間と比べ大きく変化したために正しく分類できたと考えられる．この結果から，本手法における複数の単位時間による特徴量の抽出が攻撃の検出率の向上に有効であることを確認できた．一方で，その他の攻撃については 10 秒，60 秒両方の単位時間において検出できなかった．これはパケットのヘッダに特徴の現れない攻撃であったため検出できなかったと考えられる．

4.2　MWS データセットを用いた実験

4.2.1　実験条件

MWS2017 データセットの NCD in MWSCup データセットを用いてクラスタリング及び異常検出実験を行った．文献 [9] によると，このデータセットは一般的な通信を想定したデータセットであるため，本研究の目的である実ネットワークからの異常検出に最も適すると考えられる．実験に用いる学習データとして，2014 年 10 月 22 日 10 時 04 分 33 秒の pcap ファイルから単位時間を 1 秒, 5 秒, 10 秒として特徴ベクトルを抽出した後，次元圧縮およびクラスタリングを行った．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[1] に 2017 年 7 月 18 日に取得したルールを適用して使用した．次にテストデータとして，2014 年 10 月 22 日 10 時 28 分 30 秒の pcap ファイルから同様に単位時間を 1 秒, 5 秒, 10 秒として特徴ベクトルを抽出し，クラスタリング空間に投影し異常を検出した．

4.2.2　クラスタリング実験および考察

クラスタリングの結果を表 4 に示す．1842 個の区間が

表 4　クラスタリング結果：NCD データセット

	クラス数
正常のクラス	6
異常のクラス	27
合計	33

表 5　異常クラスの例:NCD データセット

攻撃名	回数
(http_inspect) LONG HEADER	2
(http_inspect) UNKNOWN METHOD	21
(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	1
Consecutive TCP small segments exceeding threshold	3
Bad segment adjusted size ¡= 0	1
(spp_ssl) Invalid Client HELLO after Server HELLO Detected	1
COMMUNITY WEB-MISC mod_jrun overflow attempt	4

表 6　表 5 と同一クラスの他の区間:NCD データセット

攻撃名	回数
(http_inspect) LONG HEADER	2
(http_inspect) UNKNOWN METHOD	18
(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	2
Consecutive TCP small segments exceeding threshold	0
Bad segment adjusted size ¡= 0	1
(spp_ssl) Invalid Client HELLO after Server HELLO Detected	0
COMMUNITY WEB-MISC mod_jrun overflow attempt	3

33 個のクラスに分類された．また，生成されたクラスのラベルを見てみると 33 個のクラスのうち 6 個のクラスが異常を全く含まない正常クラスとなった．これら 6 個の正常クラスのうちの一つに注目し，クラスに属する別の区間を見てみるとクラスのラベルと同様に異常が含まれていなかった．また，残りの 27 個のクラスでは 1 種類以上の異常が含まれていた．

表 5 は異常を含む 27 クラスの内，あるクラスにつけられたラベルの例を示している．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 5 よりこのクラスでは 7 種類の異常が含まれていることがわかった．また，同一クラス内における他の区間に Snort を適用した結果を表 6 に示す．それぞれの検出結果を見てみると，異常の種類・数ともに類似しており，同一クラスに分類された各区間はシグネチャ型 IDS で検出した場合においても類似した結果となっていることがわかる．

4.2.3　検出実験および考察

異常検出の結果を表 7 に示す．異常検出の結果，4951 個の区間のうち既存のクラスとして認識された区間が 4645 区間となり，新たな異常として識別された区間が 306 区間となった．

正常クラスとして認識された区間を見てみるとクラスのラベルと同様に異常が含まれなかった．一方，正常クラスに分類された別の区間に Snort を適用した結果を表 8 に示す．本手法ではパケットのヘッダのみから特徴を抽出しペイロードには注目していない．これらの異常はペイロードにのみ特徴が現れるため判別できなかったと考えられる．

また，表 9 は既存のクラスではなく新たな異常として検出された区間のラベルである．この区間と最も近いクラスのラベルをみると異常のラベルが $\mathbf{0}$ ベクトルの正常クラスであったことから，異常を含む区間が新たなクラスとして認識されたと考えられる．

表 7　異常検出結果：NCD データセット

	区間数
既存のクラス	4645
新たな異常	306
合計	4951

表 8　正常クラスに分類された区間:NCD データセット

攻撃名	回数
(http_inspect) LONG HEADER	1
(http_inspect) HTTP RESPONSE GZIP DECOMPRESSION FAILED	1

表 9　新たな異常として識別された区間:NCD データセット

攻撃名	回数
(http_inspect) LONG HEADER	2
(http_inspect) UNKNOWN METHOD	25
(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	1
Reset outside window	2
ACK number is greater than prior FIN	2
COMMUNITY WEB-MISC mod_jrun overflow attempt	3

4.3　大阪府立大学の実ネットワークにおける実験

4.3.1　実験条件

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, 5 秒, 10 秒として分割し，特徴ベクトルを抽出した．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, 5 秒, 10 秒として分割し，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．

4.3.2　クラスタリング実験および考察

クラスタリングの結果，4680 個の区間が 38 個のクラスに分類され，すべてのクラスに 1 種類以上の異常が含まれていた．表 10 はあるクラスにつけられたラベルの例を示している．表 10 よりこのクラスでは 10 種類の異常が含まれていることがわかった．このクラスには 7 個の区間が含まれておりそれぞれの異常の種類を見てみると，2 種類の差異のみであったことから，同一クラスに分類された各区間は類似していることがわかる．また，表 10 で示したクラスとは別のクラスの例を表 11 に示す．表 11 よりこのクラスでは 12 種類の攻撃が含まれていることがわかった．表 10 のクラスの結果と比べると表 11 のクラスでは表 10 のクラスに含まれる攻撃以外にも，IMAP に関する異常などが含まれていた．これによって表 10 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 11 のクラスに分類された他の区間も同様に Snort の検出結果を出力するとすべての区間において共通して含まれる攻撃が

表 10　あるクラスのラベル:実ネットワーク

攻撃名	回数
(http_inspect) OVERSIZE REQUEST-URI DIRECTORY	1
(http_inspect) LONG HEADER	83
(http_inspect) UNESCAPED SPACE IN HTTP URI	5
(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	53
(http_inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE	30
TCP Timestamp is missing	74
ACK number is greater than prior FIN	14
TCP window closed before receiving data	4
Limit on number of overlapping TCP packets reached	41
SERVER-WEBAPP JBoss JMX console access attempt	1

表 11　表 10 とは異なるクラスのラベル:実ネットワーク

攻撃名	回数
(http_inspect) LONG HEADER	21
(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	51
(http_inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE	11
(spp_frag3) Fragmentation overlap	85
Consecutive TCP small segments exceeding threshold	41
TCP Timestamp is missing	4
Reset outside window	38
TCP Timestamp is outside of PAWS window	7
Bad segment adjusted size ¡= 0	2
Limit on number of overlapping TCP packets reached	1
Data sent on stream after TCP Reset sent	1
(IMAP) Unknown IMAP4 command	1

いくつかあった．その中でも“(IMAP) Unknown IMAP4 command”はデータセット全体において含まれる個数が少ないにもかかわらず，表 11 のクラスに属する区間すべてに含まれていた．また，この区間で抽出した特徴量を見てみると，宛先ポート番号 143 番パケット数等の特徴量に大きな変化が現れていた．よってこのクラスは“(IMAP) Unknown IMAP4 command”を表したクラスであると考えられる．この結果から，実ネットワークで収集したトラフィックデータからも通信の種類毎に分類することが可能であることがわかった．

4.3.3　検出実験および考察

異常検出の結果を表 12 に示す．異常検出の結果，4680 個の区間のうち既存のクラスとして認識された区間が 4513 区間となり，新たな異常として識別された区間が 167 区間となった．同一のクラスに分類された区間を表 13 と表 14 に示す．この結果を見ると異常の回数には違いはあるが，異常の種類は 2 つの区間で一致していることがわかる．この結果から，異常の種類が類似する区間を同一クラスに分類できることを確認した．

一方で，表 15 は学習データに含まれておらず，テストデータにのみ含まれる異常“(smtp) Attempted data header buffer overflow: 1016 chars ”を含むクラスのラベルである．本手法においては，この区間が新たなクラスとして識別されることが理想的であるが，実験では 1 秒，5 秒，10 秒すべての区間において既存のクラスに含まれた．これはバッファオーバーフロー等はパケットのヘッダではなく，パケットのペイロードにのみ特徴が現れる攻撃であったためと考えられる．以上のことからも，パケットのヘッダに特徴の現れない攻撃については検出が難しいことがわかる．

表 12　異常検出結果：実ネットワーク

	区間数
既存のクラス	4513
新たな異常	167
合計	4680

表 13　同一クラスに分類された区間 a:実ネットワーク

攻撃名	回数
(http_inspect) LONG HEADER	40
(http_inspect) UNKNOWN METHOD	1
(http_inspect) SIMPLE REQUEST	1
(http_inspect) UNESCAPED SPACE IN HTTP URI	5
(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	65
(http_inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE	5
(spp_frag3) Fragmentation overlap	34
Consecutive TCP small segments exceeding threshold	56
TCP Timestamp is missing	4
Reset outside window	46
ACK number is greater than prior FIN	99

表 14　同一クラスに分類された区間 b:実ネットワーク

攻撃名	回数
(http_inspect) LONG HEADER	43
(http_inspect) UNKNOWN METHOD	1
(http_inspect) SIMPLE REQUEST	1
(http_inspect) UNESCAPED SPACE IN HTTP URI	7
(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	67
(http_inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE	3
(spp_frag3) Fragmentation overlap	9
Consecutive TCP small segments exceeding threshold	44
TCP Timestamp is missing	14
Reset outside window	25
ACK number is greater than prior FIN	344

表 15　未知の異常を含む区間:実ネットワーク

攻撃名	回数
(http_inspect) LONG HEADER	61
(http_inspect) UNESCAPED SPACE IN HTTP URI	1
(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	55
(http_inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE	17
(spp_frag3) Fragmentation overlap	87
(smtp) Attempted data header buffer overflow: 1016 chars	1
Consecutive TCP small segments exceeding threshold	119
TCP Timestamp is missing	10
Reset outside window	22
TCP Timestamp is outside of PAWS window	12
Bad segment adjusted size ¡= 0	13
Limit on number of overlapping TCP packets reached	1
(POP) Unknown POP3 command	1
(POP) Unknown POP3 response	1

5.　まとめ

本稿では，パケットのヘッダから特徴ベクトルを抽出し，クラスタリングした結果にシグネチャ型 IDS でラベル付けを行うことで未知の異常の検出及び異常の判別が可能な手法を提案した． 実験では，まず予備実験として DARPA データセットを用いて本手法における複数の単位時間による特徴抽出が有効であるかを確認した．その後，実運用中のネットワークでの実験として，MWS データセットの NCD in MWSCup データセットおよび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て特徴量の抽出・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． 今後の課題として，抽出した特徴量の取捨選択やクラスタリングの際のパラメータ調整だけでなく，検出率向上のためにペイロードに含まれる特徴量の抽出などが挙げられる．

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, 竹森敬祐, 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] 今井康平, 青木茂樹, 宮本貴朗, “シグネチャ型 IDS を考慮したトラフィック特徴量のクラスタリングに基づく未知の異常検出, ”信学技報, ICSS2014-64, 2015.

[9] 高田雄太, 寺田真敏, 村上純一, 笠間貴弘, 吉岡克成, 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 Datasets 2016～, ”情処研報, 2016-CSEC-74(17), 2016.