# ペイロードの特徴量に注目した未知のネットワーク異常検出

浦川 侑之介[1,a)]　青木 茂樹[1,b)]　宮本 貴朗[1]

**概要**：侵入検知システム (IDS) に関する従来研究では，パケットのヘッダから得られる特徴のみに注目している手法が多く，ペイロードのみに特徴の表れる異常を検出することが難しかった．そこで本研究では，パケットのヘッダ情報だけでなくペイロードの特徴にも注目することにより，未知の異常を検出する手法を提案する．まず，パケットのヘッダ及びペイロードから抽出した特徴量をクラスタリングする．次に，各クラスにシグネチャ型 IDS を用いてラベル付けを行う．その後，シグネチャ型 IDS のラベルを基にして未知の異常の検出及び異常の種類の判別を行う．実験では，MWS データセット，大阪府立大学ネットワークのトラフィックに本手法を適用し，有効性を確認する．

**キーワード**：クラスタリング, ペイロード, IDS

## 1. はじめに

近年，インターネットの発達に伴いサイバー攻撃が増加傾向にある．サイバー攻撃への対策として，ネットワークに対する不正な通信を検出するための侵入検知システム (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 として，Snort[1] や Suricata[2]，The Bro[3] 等が挙げられる．シグネチャ型 IDS はあらかじめ異常な通信をパターンファイルに記録しておき，観測している通信と一致するパターンがパターンファイルに存在すれば異常を検知する IDS である．しかし，パターンファイルに定義されていない異常については亜種を含め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，機械学習を用い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 [5,6,7,8] の手法が挙げられる．アノマリ型 IDS はあらかじめ正常なパターンを学習しておき，学習した正常パターンからの外れ値を異常として検出する IDS 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種類の異常であるか判別できないという問題点がある．また，実運用中のネットワークにおける通信には正常データのみでなく異常も含まれている可能性があるため，正常パターンを定義するデータを用意することが難しいといった問題点もある．

文献 [9,10] では，学習したクラスに対して異常のラベルを付加することにより異常の種類を識別する手法を提案している．文献 [9] では，トラフィックデータから抽出した特徴ベクトルをクラスタリングし，学習することで異常を検知する手法を提案している．学習したクラスにユーザがラベルをつけることによりどのような異常が含まれているかの判別を可能にしている．また文献 [10] 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加することで，アノマリ型 IDS では不可能であった異常の種類の提示に成功している．しかし，この手法ではパケットのヘッダから抽出した特徴量のみを用いているため，ペイロードのみに特徴が表れる異常の検知が難しかった．

また，文献 [11] ではペイロードから抽出した特徴量を用いてマルウェアの感染検知を行い，ペイロードから抽出した特徴量の中でどの特徴量が有効であるかの評価をしている．利用している特徴量は，ASCII 文字コードの出現頻度や特徴的な文字列の出現頻度，HTTP リクエスト長であ

[1] 大阪府立大学
Osaka Prefecture University
[a)] sya01040@edu.osakafu-u.ac.jp
[b)] aoki@kis.osakafu-u.ac.jp

る．実験では，マルウェアの種類ごとに検出に有効な特徴量があることを確認している．

そこで本稿では，ヘッダから抽出した特徴量にペイロードに出現する ASCII 文字列の種類数と長さに基づく特徴量を加えて異常を検出する手法を提案する．

## 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 の代表的な手法である文献 [5,6,7,8] と学習したクラスに対して異常のラベルを付加することにより異常の種類を識別する手法である文献 [9,10] について述べる．文献 [5] では，パケットのエントロピーによる異常検出手法が提案されている．この手法では 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異常として検出している．

文献 [6] では，ネットワーク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を検出する手法を提案している．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する．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を識別している．

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抽出した特徴量に対してクラスタリングを行い学習することによりマルウェアの通信を識別する手法を提案している．特徴量として，一般的な特徴量 70 種類とマルウェア特有の特徴量 25 種類を用いている．DBSCAN を用いてクラスタリングし，クラス内のデータと検出したい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に分類され既知のマルウェアとする．閾値以上であった場合は未知のマルウェアとする．

文献 [5,6,7,8] 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異常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 [9,10] では，学習したクラスに対して異常のラベルを付加することにより異常の種類を識別する手法を提案している．

文献 [9] では，トラフィックデータから抽出した特徴ベクトルに対して，コーシー確率密度関数を用いてクラスタリングし，学習を行うことにより異常を検知する手法を提案している．コーシー密度関数を用いることにより，リアルタイムで学習モデルを更新しながら異常を検知することを可能にしている．また，学習したクラスにユーザーがラベルをつけることによりどのような異常が含まれているかの判別を可能にしている．

文献 [10] 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加することで，アノマリ型 IDS ではできなかった異常の識別を可能としている．

文献 [10] の手法では，パケットのヘッダのみから特徴量を抽出しているためペイロードのみに特徴が表れる異常を検出できないことが課題となっていた．そこで本稿では，ヘッダから抽出した特徴量にペイロードに出現する文字列に注目した特徴量を加えて異常を検出する手法を提案する．

## 3. 提案手法

本手法の異常検出の手順について述べる．手法の流れを図 1 に示す．本手法は学習と異常検出の 2 つの処理に分かれている．学習時の処理では，学習データとなるトラフィックデータを一定の単位時間で分割し，各区間に含まれるパケットのヘッダとペイロードから 75 次元の特徴ベクトルを抽出する．75 次元の特徴ベクトルではクラスタリングをする際に次元数が大きすぎるため, 主成分分析法 (PCA) を用いて次元を圧縮する．その後，特徴ベクトルをクラスタリングする．そして，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をそのクラスの代表ベクトルとし，シグネチャ型 IDS を用いてクラスに含まれている異常の種類を特定し，ラベルとする．

異常検出時の処理では，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の抽出，次元圧縮を行い，学習した空間に特徴ベクトルを投影する．そして，投影された特徴ベクトルと最も近いクラスの代表ベクトルとの距離を算出し，ラベルを出力すること

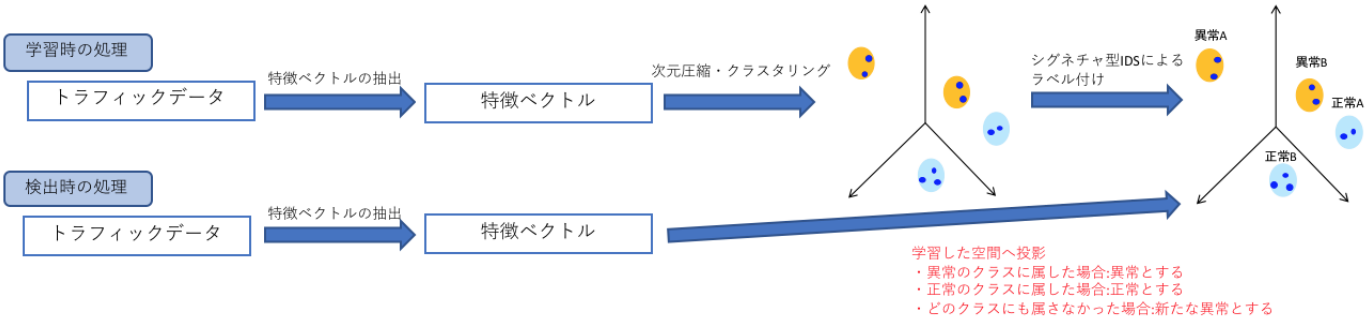


**図 1**　提案手法の概要
**Fig. 1**　Outline of Proposed Method.

により異常の種類を特定する．

### 3.1　特徴量の抽出

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，単位時間 $\omega$ で分割したものを区間 $t$ とし，$N$ 個収集する．各区間に含まれるパケットのヘッダから表 1 に示すパケット数や宛先ポート番号種類数などの 53 種類の特徴量を抽出する．パケットのヘッダの特徴量に着目することにより，トラフィック量やポートの種類数などの統計的な情報に変化が現れる異常を識別可能である．さらに，ペイロードから特徴量を抽出する．標的型攻撃におけるビーコン通信などでは，類似する文字列を多く含んでいる通信を多く行う可能性が高いと考えられる．そのため，本手法ではペイロードから抽出する特徴量として，ASCII 文字列の種類数と ASCII 文字列の長さに基づく特徴量を抽出する．文字列の種類数に基づく特徴抽出の概要を図 2 に示す．まず，各区間に含まれているパケットのペイロードに出現する文字列をすべて抽出する．図の例では，「GET」「1」「HTTP/1.1」「HTTP/1.1」「200」「OK」が抽出される．次に，文字列の出現回数ごとに種類数を求める．図の例では，出現回数 1 回の ASCII 文字列が「GET」「1」「200」「OK」であるため種類数は 4，出現回数が 2 回の文字列は種類数が 1 となる．そして，出現回数が 1~10 回であった文字列の種類数を足し合わせることにより，出現回数が 1~10 回の文字列の種類数を特徴として抽出する．11~20 回以降も同様に算出することにより文字列の種類数に基づく特徴を 11 種類抽出する．また，文字列の長さに基づく特徴抽出の概要を図 3 に示す．まず，文字列の種類数に基づく特徴抽出の場合と同様に各区間に含まれているパケットのペイロードに出現する文字列をすべて抽出する．次に，文字列の長さごとに出現回数をカウントする．図の例では，文字列長が 1 の文字列は「1」1 つだけであり，文字列長が 3 の文字列は「200」「GET」の 2 種類となる．同様の処理により全ての文字列長について種類数を調べる．そして，文字列長が 1~10 であった文字列

**表 1**　ヘッダから抽出した特徴量の一覧
**Table 1**　List of Packet Header Features.

| | |
|---|---|
| パケットサイズ平均 | 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 | |

の出現回数を足し合わせることにより，文字列長が 1~10 の文字列の数を特徴として抽出する．また，11~20 以降も同様に計測することにより文字列の長さに基づく特徴を 11 種類抽出する．ペイロードから抽出する 22 種類の特徴量を表 2 に示す．表 1 と表 2 の特徴量を合わせて特徴ベクトル $\boldsymbol{I}_t$ とする．

### 3.2　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，類似しない異常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

表 2　ペイロードから抽出した特徴量の一覧
**Table 2**　List of Packet Payload Features.

| | |
|---|---|
| 出現回数が 1~10 回の文字列の種類数 | 出現回数が 11~20 回の文字列の種類数 |
| 出現回数が 21~30 回の文字列の種類数 | 出現回数が 31~40 回の文字列の種類数 |
| 出現回数が 41~50 回の文字列の種類数 | 出現回数が 51~60 回の文字列の種類数 |
| 出現回数が 61~70 回の文字列の種類数 | 出現回数が 71~80 回の文字列の種類数 |
| 出現回数が 81~90 回の文字列の種類数 | 出現回数が 91~100 回の文字列の種類数 |
| 出現回数が 101 回以上の文字列の種類数 | 文字列長が 1~10 の文字列の数 |
| 文字列長が 11~20 の文字列の数 | 文字列長が 21~30 の文字列の数 |
| 文字列長が 31~40 の文字列の数 | 文字列長が 41~50 の文字列の数 |
| 文字列長が 51~60 の文字列の数 | 文字列長が 61~70 の文字列の数 |
| 文字列長が 71~80 の文字列の数 | 文字列長が 81~90 の文字列の数 |
| 文字列長が 91~100 の文字列の数 | 文字列長が 101 以上の文字列の数 |

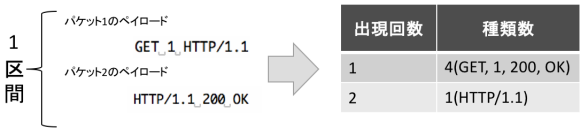


図 2　文字列の種類数の抽出例
**Fig. 2**　Extracting of Number of String Types from Packet Payload.

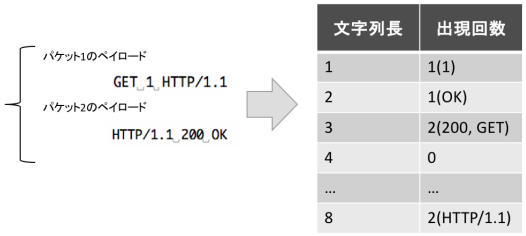


図 3　文字列の長さの抽出例
**Fig. 3**　Extracting of String Length from Packet Payload.

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．

そこで特徴ベクトルをクラスタリングする．クラスタリングする際に特徴ベクトルが 75 次元では大きすぎるため主成分分析法を用い次元を圧縮する．ここでは累積寄与率が 80%以上となる最小の次元数で圧縮する．次元圧縮した空間を $\boldsymbol{S}$ 空間と呼ぶと，$\boldsymbol{S}$ 空間上での座標値は，$\boldsymbol{I}_t = (s_{t1}, s_{t2}, \cdots, s_{tr})$ で表される．その後，$\boldsymbol{S}$ 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．この手法はあらかじめ分類するクラス数を指定する必要がないため，今回のような異常の種類数があらかじめわから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

前節までで述べた方法により異常検出のためのアノマリ型 IDS を構築した．しかし，このアノマリ型 IDS では異常を検出できるが，どのような異常を含んでいるかは識別できない．そこで，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加する．それにより，付加したラベル情報により各クラスがどのような異常を含むのかを識別できるようにする．

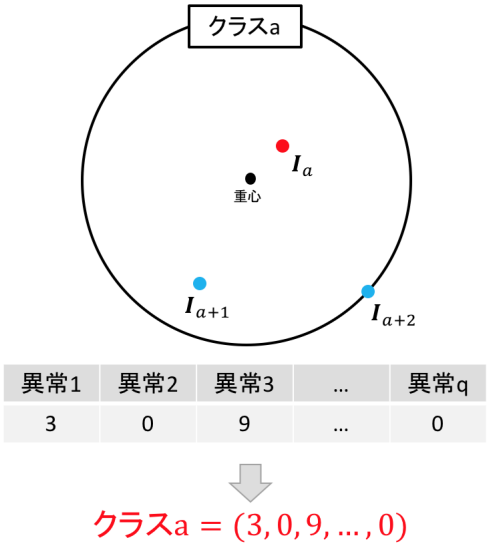


図 4　クラスへのラベル付けの例
**Fig. 4**　Example of Labeling.

ラベル付けの概要を図 4 に示す．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．出力された結果をクラスにラベルとして付加することで，そのクラスにどのような異常が含まれているのかを表す．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，$(s'_{t1}, s'_{t2}, \cdots, s'_{tv}, \cdots, s'_{tq})$ をラベルとして付加する．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．すべてのクラスに対してラベルを付加することによって未知の異常検出および異常の識別が可能な IDS を構成することができる．また，同一のラベルが複数のクラスに付加された場合はそれぞれ別のクラスとして扱う．

### 3.4　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 $\omega$ で分割しそれぞれの区間から 75 次元の特徴ベクトルを抽出する．その後，75 次元の特徴ベクトルを学習時と同様に低次元に圧縮し，学習した $\boldsymbol{S}$ 空間に投影する．図 5 に示すようにクラス $a$ とクラス $b$ が存在するとき，新たな特徴ベクトル $\boldsymbol{I}_t$ が，各クラス重心との距離においてクラス $a$ の重心との距離が最も小さいと算出されたとする．特徴ベクトル $\boldsymbol{I}_t$ とクラス $a$ の重心との距離を $e$ とする．また，クラス $a$ 内の特徴ベクトル $\boldsymbol{I}_a, \boldsymbol{I}_{a+1}, \boldsymbol{I}_{a+2}$ の内，重心との距離が最も大きい特徴ベクトルである $\boldsymbol{I}_{a+2}$ と重心との距離を $f$ とする．$e \leq f$ が成り立つため，特徴ベクトル $\boldsymbol{I}_t$ はクラス $a$ に属すると判断する．そしてクラス $a$ に付加したラベルを識別結果として出力する．次に，新たな特徴ベクトル $\boldsymbol{I}_{t+1}$ は，各クラス重心との距離においてクラス $b$ との距離が最

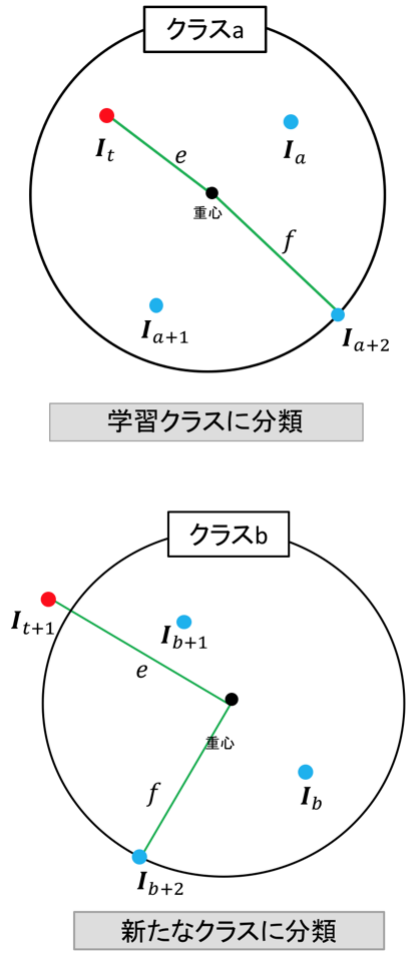


図 5　異常検出の例
**Fig. 5**　Example of Anomaly Detection.

も小さいと算出されたとする．特徴ベクトル $\boldsymbol{I}_{t+1}$ とクラス $b$ の重心との距離を $e$ とする．クラス $b$ 内の特徴ベクトル $\boldsymbol{I}_b, \boldsymbol{I}_{b+1}, \boldsymbol{I}_{b+2}$ の内，重心との距離が最も大きい特徴ベクトルである $\boldsymbol{I}_{b+2}$ と重心との距離を $f$ とすると，$e > f$ が成り立つため，特徴ベクトル $\boldsymbol{I}_{t+1}$ はクラス $b$ には属さないと判断し，新たな異常と識別する．

## 4.　実験

本手法の有効性を確認するために MWS2018 データセットおよび大阪府立大学のネットワークで収集したトラフィックデータを用いて実験を行った．クラスに対するラベル付けにはシグネチャ型 IDS の一つである Suricata を用いた．

### 4.1　MWS2018 データセットにおける実験

正常な通信のみを学習し，異常を含む通信中の異常を検出できることを確認するために実験を行った．学習データ及びテストデータとしては，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．このデータセットはマルウェアを実行したホストの通信をチャプチャしたものである．また，BOS データセットのデータにはマルウェアの進行度が示されており，マルウェアにより通信が発生したか，またどのように通信が行われたデータであるかが示されている．実験には，学習データとしてマルウェアは実行したが通信は発生しなかった進行度 2 の 2017 年 8 月 17 日に収集された 24 時間のデータを正常のみのデータとして用いた．学習データを用いて特徴量の抽出及びクラスタリングとクラスタリング結果の各クラスに対するシグネチャ型 IDS を用いたラベル付けを行った．テストデータとしてはマルウェアを実行し，マルウェアの通信を継続的に観測できた進行度 8 の 2018 年 1 月 23 日に収集された 24 時間のデータを異常を含むデータとして用いた．学習データ，テストデータともに特徴量抽出の際の単位時間は 30 秒とし 2880 区間に分割した．また，マルウェアの通信相手を検出対象である異常な通信とすると，検出対象の通信を含む区間が 259 区間あった．これらのデータセットを用いて本手法における検出結果とヘッダのみから抽出した特徴ベクトルを用いた場合 (従来手法 [10]) の検出結果との比較を行った．

#### 4.1.1　クラスタリング結果・考察

ヘッダ及びペイロードから抽出した特徴ベクトルに対してクラスタリングを行った結果，2880 個の区間が 51 個のクラスに分類された．各クラスの代表ベクトルに対して Suricata を適用して異常検出を行うと，異常が含まれる区間は存在しなかった．学習データに含まれている通信ではマルウェアによる通信が発生せず，異常な通信が存在しなかったため全てのクラスで異常を検出しなかったと考えられる．そこで，51 クラス全てを正常クラスとして扱う．また，ヘッダのみから抽出した特徴ベクトルを用いてクラスタリングした結果についても 51 クラスとなった．同様に，各クラスの代表ベクトルに Suricata により異常検出を行うと，異常が含まれる区間は存在しなかった．

#### 4.1.2　検出実験結果・考察

テストデータにおける検出実験の結果を表 3 に示す．2880 区間の内，2636 区間を学習時のクラスに分類した．一方，244 区間を新たなクラス (異常) であると分類した．また，検出対象の通信が含まれている 259 区間中 175 区間を学習時の正常クラス，84 区間を学習した正常クラスではなく新たなクラスに分類した．また，検出対象の通信が含まれていない 2621 区間中 2461 区間を学習時の正常クラス，160 区間を新たなクラスに分類した．また，検出対象の通信が含まれている 259 区間について，1 区間に含まれる検出対象のパケット数の影響を調べた結果を表 4 に示す．表中，1 列目に 1 区間中に含まれる検出対象のパケット数を示し，2 列目に新たなクラス (異常) として検出できた数，3 列目に学習時のクラス (正常) として識別した数を示している．検出対象の通信が 1 区間に 80 パケット以上含まれ

表 3　本手法の検出結果 (MWS2018 Dataset)
**Table 3**　Detection Result of Proposed Method(MWS2018 Dataset).

| 検出結果 | 分類された総区間数 | 異常な通信を含む | 異常な通信を含まない |
|---|---|---|---|
| 学習時のクラス | 2636 区間 | 175 区間 | 2461 区間 |
| 新たなクラス | 244 区間 | 84 区間 | 160 区間 |
| 合計 | 2880 区間 | 259 区間 | 2621 区間 |

ていた場合は 100%の精度で新たなクラスに分類できていた．一方で，1 区間に含まれるパケット数が 60 パケット未満の場合は 184 区間中 174 区間 (94.5%) と多くの検出対象の通信が含まれる区間を学習時の正常なクラスに分類していた．また，1 区間に含まれる検出対象のパケットの割合にも着目したところ，区間の 20%以上が検出対象のパケットの場合に 100%の精度で新たなクラスに分類できていた．一方，検出対象のパケットが区間の 20%未満の場合は 8.4%であった．このことから，検出対象の通信がある程度多ければ正常な区間との差異が生まれ，新たなクラスに分類できることを確認できた．一方で，検出対象の通信が少なく正常な区間との違いが表れなかった場合は学習時のクラスに分類されていた．また，検出対象の通信が含まれていない 160 区間を新たなクラスに分類した．これは，学習時のクラスに類似した特徴を持ったクラスがなかったためであると考えられる．学習データを増やし，クラス数を増やすことにより改善できると考えられる．

また，テストデータの 2880 区間全てに Suricata を用いて異常検出を行ったところ，9 区間に POLICY HTTP traffic on port 443 (POST) という異常が含まれていた．また，この 9 区間は検出対象の通信が含まれていた区間であった．Suricata では，外部の異常なホストが含まれていた 259 区間中 9 区間を検出できたが，本手法では 259 区間中 84 区間を新たなクラスに分類できていた．このことから，本手法を用いることでシグネチャ型 IDS では検出できない異常が含まれている場合においても新たなクラスに分類され，検出可能であることを確認できた．

また，従来手法 [10] に基づきヘッダのみから得られた特徴ベクトルを用い検出実験を行った結果，本手法と同様に検出対象の通信が多い区間を正しく識別することができていた．また，本手法と比較したところ検出結果に大きな差異は見られなかった．これは，本実験で用いたデータセットに含まれる異常がヘッダから抽出した特徴量にも特徴が現れる異常であったためであると考えられる．今後，様々なデータセットに対して実験を行い，ペイロードから抽出した特徴量の有効性について確認する予定である．

### 4.2　大阪府立大学のトラフィックデータにおける実験

本手法の実運用中のネットワークでの有効性を確認するために実験を行った．実験では，学習データとして，大阪府立大学のネットワークとインターネットの間にあるファイアウォールの外側で 2017 年 11 月 16 日に収集した 1 時間のトラフィックデータを使用した．学習データを用いて特徴量の抽出及びクラスタリングとクラスタリング結果の各クラスに対するシグネチャ型 IDS を用いたラベル付けを行った．次にテストデータとして 2017 年 11 月 17 日に収集した 1 時間のトラフィックデータを使用した．パケットの収集方法は，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した．また，学習データ，テストデータともに特徴量抽出の際の単位時間は 1 秒とし，3600 区間に分割した．

表 4　異常な通信の識別率 (MWS2018 Dataset)
**Table 4**　Detection Rate of Unusual(MWS2018 Dataset).

| 1 区間中の異常なホストとのパケット数 | 新たなクラス | 学習時のクラス |
|---|---|---|
| 100 パケット以上 | 58 区間 | 0 区間 |
| 80 パケット以上 100 パケット未満 | 13 区間 | 0 区間 |
| 60 パケット以上 80 パケット未満 | 3 区間 | 1 区間 |
| 60 パケット未満 | 10 区間 | 174 区間 |

表 5　あるクラス a のラベル
**Table 5**　Label of Cluster “a”.

| 異常名 | 回数 |
|---|---|
| SCAN Sipvicious Scan | 1 |
| POLICY Suspicious inbound to MSSQL port 1433 | 20 |
| POLICY Suspicious inbound to Oracle SQL port 1521 | 2 |
| SCAN Sipvicious User-Agent Detected (friendly-scanner) | 5 |
| POLICY Protocol 41 IPv6 encapsulation potential 6in4 IPv6 tunnel active | 1 |
| DNS Non-DNS or Non-Compliant DNS traffic on DNS port Reserved Bit Set | 0 |
| CINS Active Threat Intelligence Poor Reputation IP group 81 | 1 |

表 6　表 5 と同一クラスの区間のラベル
**Table 6**　Label of Section in Table 5 Cluster.

| 異常名 | 回数 |
|---|---|
| SCAN Sipvicious Scan | 1 |
| POLICY Suspicious inbound to MSSQL port 1433 | 27 |
| POLICY Suspicious inbound to Oracle SQL port 1521 | 4 |
| SCAN Sipvicious User-Agent Detected (friendly-scanner) | 5 |
| POLICY Protocol 41 IPv6 encapsulation potential 6in4 IPv6 tunnel active | 1 |
| DNS Non-DNS or Non-Compliant DNS traffic on DNS port Reserved Bit Set | 1 |
| CINS Active Threat Intelligence Poor Reputation IP group 81 | 1 |

#### 4.2.1　クラスタリング結果

クラスタリングを行った結果，3600 個の特徴ベクトルが 102 個のクラスに分類された．また，すべての区間に 1 種類以上の異常が含まれていた．表 5 はあるクラス a のラベルである．また，表 6 は表 5 と同一のクラスに分類された他の区間のシグネチャ型 IDS の検出結果である．表 5 と表 6 に示す結果から，同一のクラスに分類された区間では含まれている異常の種類及び回数が類似していることを確認した．また，表 7 は表 5 のクラスと距離が離れたクラスに属するある区間におけるシグネチャ型 IDS の検出結果である．表 5 と表 7 に示す結果より，距離の離れたクラスでは含まれている異常の種類が異なっていることを確認した．これらの結果から，特徴ベクトルのクラスタリングにより類似する異常を正しく分類できることを確認した．

#### 4.2.2　検出実験結果

テストデータを用いて検出実験を行った．テストデー

表 7　表 5 と距離が離れたクラスのラベル
**Table 7**　Label of Distant Cluster from Table 5 Cluster.

| 異常名 | 回数 |
|---|---|
| SCAN Potential SSH Scan | 1 |
| POLICY Suspicious inbound to MSSQL port 1433 | 23 |
| POLICY Suspicious inbound to Oracle SQL port 1521 | 2 |
| DOS Possible SSDP Amplification Scan in Progress | 1 |
| POLICY Outdated Flash Version M2 | 1 |
| GPL SNMP public access udp | 5 |

表 8　テストデータの検出結果
**Table 8**　Detection Result of Test Data.

| 検出結果 | 分類された区間数 |
|---|---|
| 学習時のクラス | 2812 区間 |
| 新たな異常 | 788 区間 |

タにおける検出実験の結果を表 8 に示す．3600 区間の内 2812 区間を学習時のクラスに分類した．一方，788 区間を新たな異常であると識別した．

表 9 は学習時のあるクラス b のラベルである．また，表 10 はテストデータにおける表 9 のクラスに分類された区間のシグネチャ型 IDS の検出結果である．表 9 と表 10 に示す結果より，異常検出処理においても，クラスのラベルとそのクラスに分類された区間では異常名や回数が類似していることを確認した．

本手法とシグネチャ型 IDS の Suricata における検出結果にどのような差異があるかを確認するために検出結果の比較を行った．学習データとテストデータの各区間に対して Suricata を用いた際の検出結果を表 11 に示す．テストデータにおいて学習データに含まれている異常のみで構成されている区間は 2649 区間あった．また，学習データに含まれていない新たな異常を含む区間は 951 区間あった．本手法と Suricata における検出結果の比較を表 12 に示す．Suricata において学習データに含まれていない異常が含まれているとされた 951 区間の内，204 区間を新たなクラスに分類し，747 区間を既存のクラスに分類した．また，Suricata において学習データに含まれている異常のみで構成されている 2649 区間の内，584 区間を新たなクラスに分類し，2065 区間を既存のクラスに分類した．

本手法で用いたペイロードの特徴量の検出結果に対する影響を確認するため，本手法とヘッダのみの特徴量を用いた従来手法 [10] との比較を行った．Suricata において学習データにない未知の異常が含まれているとされた 951 区間の分類結果の比較を表 13 に示す．また，Suricata において学習データにある既知の異常のみで構成された 2649 区間の分類結果の比較を表 14 に示す．学習データに含まれていない新たな異常が含まれている区間，学習データにある異常のみで構成された区間ともに従来手法に比べ本手法の方が新たなクラスに分類されている区間が多かった．

表 9　あるクラス b のラベル
**Table 9**　Label of Cluster “b”.

| 異常名 | 回数 |
|---|---|
| SCAN Sipvicious Scan | 1 |
| POLICY Suspicious inbound to MSSQL port 1433 | 15 |
| SCAN Sipvicious User-Agent Detected (friendly-scanner) | 5 |
| POLICY Protocol 41 IPv6 encapsulation potential 6in4 IPv6 tunnel active | 1 |
| SNMP public access udp | 2 |
| CINS Active Threat Intelligence Poor Reputation IP group 20 | 1 |

表 10　表 9 のクラスに分類された区間のラベル
**Table 10**　Label of Classified Section in Table 9 Cluster.

| 異常名 | 回数 |
|---|---|
| SCAN Sipvicious Scan | 1 |
| POLICY Suspicious inbound to MSSQL port 1433 | 21 |
| SCAN Sipvicious User-Agent Detected (friendly-scanner) | 5 |
| POLICY Protocol 41 IPv6 encapsulation potential 6in4 IPv6 tunnel active | 1 |
| SNMP public access udp | 0 |
| CINS Active Threat Intelligence Poor Reputation IP group 20 | 1 |

表 11　Suricata による検出結果
**Table 11**　Result of Suricata Detection.

| Suricata による検出結果 | 分類された区間数 |
|---|---|
| 学習データに含まれている異常のみ | 2649 区間 |
| 学習データに含まれていない新たな異常を含む | 951 区間 |

表 12　本手法と Suricata の検出結果の比較
**Table 12**　Comparison of Detection Result Proposed Method and Suricata.

| | | Suricata | |
|---|---|---|---|
| | | 未知の異常 | 既知の異常 |
| 本手法 | 新たなクラス | 204 | 584 |
| | 既存のクラス | 747 | 2065 |

表 13　本手法と従来手法の検出結果の比較 (未知の異常を含む区間)
**Table 13**　Comparison of Proposed Method and Conventional Method (Section of unknown attacks).

| | | 従来手法 | |
|---|---|---|---|
| | | 新たなクラス | 既存のクラス |
| 本手法 | 新たなクラス | 55 | 149 |
| | 既存のクラス | 10 | 737 |

表 14　本手法と従来手法の検出結果の比較 (既知の異常のみの区間)
**Table 14**　Comparison of Proposed Method and Conventional Method (Section of known attacks).

| | | 従来手法 | |
|---|---|---|---|
| | | 新たなクラス | 既存のクラス |
| 本手法 | 新たなクラス | 127 | 457 |
| | 既存のクラス | 34 | 2031 |

### 4.2.3　クラスタリング実験結果の考察

トラフィックデータを単位時間で分割し，各区間から 75 次元の特徴ベクトルを抽出した．特徴ベクトルをクラスタリングすることによりアノマリ型 IDS を構築した．

学習データから抽出した特徴ベクトルに対してクラスタリングを行った結果，類似する異常を含む区間を同一のクラスに分類できることを確認した．また，異なる異常を含

む区間については別のクラスに分類されていることも確認した．この結果より，今回抽出した特徴量によって異常を分類することができたと考えられる．

#### 4.2.4 検出実験結果の考察

テストデータから特徴ベクトルを抽出し，学習した空間への投影を行った．その後，各クラスの重心との距離が最も近いクラスを選択して，クラスの重心との距離を算出した．また，クラス内で重心から最も遠い特徴ベクトルと重心との距離を算出し，先ほどの距離と比較することで学習データに含まれていた異常か新たな異常かを識別した．

本手法における検出結果と Suricata における検出結果の違いについて考察する．Suricata のようなシグネチャ型 IDS では攻撃の亜種などの類似する異常を別の異常として検出する．一方，本手法においてはパケットから抽出した特徴量に基づき類似する異常を同一のクラスに分類する．そのため，新たな異常として識別される区間が少なくなったと考えられる．Suricata の検出結果を確認すると学習データには存在していなかった“SCAN Potential VNC Scan 5900-5920”というスキャン攻撃を含む区間を既存のクラスに分類していた．この区間が分類されたクラスのラベルを確認してみると類似する他のスキャン攻撃が存在した．そのため，特徴量が類似し学習時のクラスに分類されたと考えられる．この結果より，学習データに現れなかった新たな異常についても類似するクラスに分類できることを確認できた．以上のことから，学習時に観測されなかった未知の異常についてもクラスのラベルを確認することで異常の種類を識別可能であると考えられる．

本手法と従来手法の検出結果の違い及びペイロードからの特徴抽出について考察する．実験の結果，従来手法に比べ本手法の方が新たなクラスとして検出される区間が多くなったことを確認できた．これは，ペイロードから抽出した特徴量を増やしたことにより詳細に分類され，新たなクラスとして識別された区間が従来手法よりも増加したと考えられる．また，ペイロードから抽出した特徴量を用いることにより，ヘッダのみの特徴量を用いた従来手法との検出結果の違いを確認できたが，実際に本手法によりどのような異常が識別可能になったのかを確認できなかった．今後，あらかじめ異常のラベルがついたデータセットを用いることにより本手法において新たにどのような異常が識別可能になったのかを検証する必要がある．

## 5. まとめ

本稿では，パケットから特徴ベクトルを抽出し構築したアノマリ型 IDS とシグネチャ型 IDS の検出結果を組み合わせる事により未知の異常検出及び異常の種類の判別が可能な手法を提案した．また，従来手法では用いていなかったペイロードから抽出した特徴量を用いることによりペイロードのみに特徴が表れる異常の識別を図った．MWS データセットを用いた実験では，シグネチャ型 IDS だけでは検出困難な異常についても検知できることを示した．また，大阪府立大学におけるトラフィックデータを用いた実験では，ヘッダに加えペイロードからも特徴量を抽出することによりヘッダのみの特徴量を用いた場合と比べ検出結果に違いが生じたことを確認した．

今後の課題としてはペイロードから抽出する特徴量の再考や，実際にペイロードから抽出した特徴量を加える事によりどのような異常を新たに識別可能になったのかなど実験結果を精査することなどが挙げられる．

### 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，信学技報，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，信学技報, 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，信学技報, NS2004-144, pp.45-48(2004).

[8] Hatada, Mitsuhiro, 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] Škrjanc, I., Ozawa, S., Ban, T., and Dovžan, D.:Large-scale cyber attacks monitoring using Evolving Cauchy Possibilistic Clustering, Applied Soft Computing, 62, pp.592-601(2018).

[10] 谷澤俊樹, 青木茂樹, 宮本貴朗：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，Computer Security Symposium 2017(CSS2017) 講演論文集, pp.1397-1403(2017).

[11] 大月優輔，市野将嗣，川元研治，畑田充弘，吉浦裕：マルウェア感染検知のためのトラヒックデータにおけるペイロード情報の特徴量評価，Computer Security Symposium 2012(CSS2012) 講演論文集, pp.691-698(2012).

[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～，情報処理学会論文誌, Vol.2018-CSEC-82，No.38，pp.1-8(2018).