LSTMを用いた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：本研究では，パケットのヘッダから単位時間ごとに抽出した特徴ベクトルを学習することによりネットワークの異常を検出する手法を提案する．単位時間ごとに特徴を抽出することで特徴抽出の処理にかかるコストを削減し，トラフィック量の多い大規模ネットワークに適用可能な異常検出手法の実現を目指す．学習には，画像処理や自然言語処理等の分野で高い性能を示している Deep Learning を使用する．ここでは，時系列データの処理に適しているリカレントニューラルネットワークの拡張である LSTM（Long short-term memory）を用いる．まず，学習期間中のトラフィックデータから特徴ベクトルを抽出し，LSTM で学習する．その後，別の期間に収集したトラフィックデータからも同様に特徴ベクトルを抽出する．抽出した特徴ベクトルに対して学習した LSTM を用いることにより，異常の発生を検出する．実験では，MWS データセットおよび大阪府立大学ネットワークのトラフィックデータに本手法を適用し，有効性を確認した．

キーワード：異常検出, Deep Learning, LSTM

Anomaly Detection of Network using LSTM

Yunosuke Urakawa[1,a)]　Shigeki Aoki[1,b)]　Takao Miyamoto[1]

Abstract: In this research, we propose a method to detect anomalies on a network by learning feature vectors extracted from packet headers at every unit time. We reduce cost of feature extraction processing and aim to realize the anomaly detection method applicable to large-scale networks with large traffic volume by using extracted features at every unit time. We use Deep Learning which shows high performance in the field of image processing and natural language processing. Here, we use LSTM (Long short-term memory) which is an extension of recurrent neural networks suitable for processing time-series data. First, feature vectors are extracted from traffic data during learning period and learned by using LSTM. After that, we extract feature vectors from traffic data in another period and identify anomalous communication and normal communication by using learned LSTM. In the experiment, the method was applied to MWS dataset and traffic data of Osaka Prefecture University network. We confirmed the effectiveness our approach.

Keywords: Intrusion Detection, Deep Learning, LSTM

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] 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

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) sya01040@edu.osakafu-u.ac.jp
b) aoki@kis.osakafu-u.ac.jp

© 2019 Information Processing Society of Japan

点がある．文献 [3] 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 [4], [5], [6] 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，アノマリ型 IDS ではシグネチャ型 IDS の問題点であった未知の異常の検出は可能であるが，シグネチャ型 IDS に比べ検出精度が低い．アノマリ型 IDS の検出精度を向上させるために機械学習を用いた様々な手法が提案されている．文献 [7], [8], [9] では，トラフィックから抽出した特徴ベクトルにクラスタリングを行い異常を検出する手法を提案している．

近年の画像認識や自然言語処理等の分野で Deep Learning を用いた手法が注目を集めている．さらに，Deep Learning をネットワークの異常検知に応用している研究もある．文献 [10] では，CNN（Convolutional Neural Network）と RNN（Recurrent Neural Network）の拡張である LSTM（Long short-term memory）を用いたネットワークの異常検出手法を提案している．また，文献 [11] では RNN の拡張である GRU（Gated Recurrent Unit）と多層パーセプトロンを組み合わせた異常検出手法を提案している．文献 [11] では実験の際にセッションごとに抽出した特徴ベクトルを使用しているため，大規模ネットワークに適用した際に特徴抽出処理に時間がかかる可能性がある．

本稿では，パケットのヘッダから単位時間ごとに抽出した特徴ベクトルを LSTM で学習することにより異常を検出する手法を提案する．本手法は，パケットのヘッダから単位時間ごとに特徴を抽出することで特徴抽出の処理にかかるコストを削減し，トラフィック量の多い大規模ネットワークに適用できるようにすることを目的とした．また，実験では MWS データセットおよび大阪府立大学ネットワークのトラフィックデータに本手法を適用し，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 [4], [5], [6] と機械学習を用いた異常検出手法の文献 [7], [8], [9] および Deep Learning を用いた異常検出手法の文献 [10], [11] について述べる．文献 [4] 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 [5] 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 [6] では，複数の特徴ベクトル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ベクトル，フロー毎のフラグの出現回数を数値化した特徴ベクトル，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ベクトル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ベクトルを抽出し，学習用データの値と閾値以上離れている特徴ベクトルが存在する場合に異常であると判断する．文献 [4], [5], [6] の手法のようなアノマリ型 IDS ではシグネチャ型 IDS の問題点であった未知の異常の検出は可能であるが，シグネチャ型 IDS に比べ検出精度が低い．

アノマリ型 IDS の検出精度を向上させるために機械学習を用いた手法が提案されている．文献 [7] では，パケットトレース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．特徴ベクトルとして，一般的な特徴ベクトル 70 種類とマルウェア特有の特徴ベクトル 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 [8] 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利用し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 [9] では，パケットのヘッダ情報から抽出した特徴ベクトル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

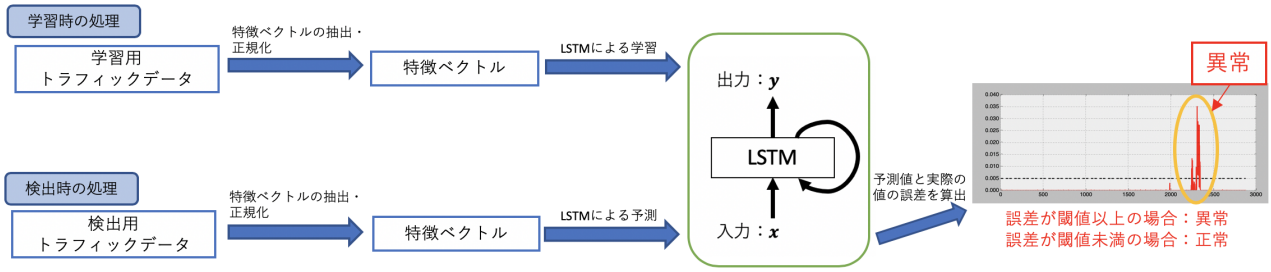

図 1　提案手法の概要
Fig. 1　Outline of Proposed Method.

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ベクトルを抽出し，主成分分析によって特徴ベクトル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．

従来の機械学習を用いた手法よりもさらに検出精度を向上させるため Deep Learning を用いた手法が提案されている．文献 [10] では，トラフィックの特徴を CNN で学習し異常を検出する手法と RNN の拡張である LSTM で学習し異常を検出する手法の 2 種類の手法を提案している．CNN での学習ではパケットのペイロードから抽出した特徴ベクトルを用いて学習および異常の検出をしている．一方，LSTM での学習ではペイロードの文字列を用いて学習および異常の検出をしている．実験では，CNN よりも LSTM を用いた際の検出率が高くなったと述べられている．文献 [11] では，RNN の拡張である GRU と多層パーセプトロンを組み合わせて学習することで異常を検出する手法を提案している．GRU で学習した後，多層パーセプトロンでさらに学習することでパケットから抽出した特徴ベクトルを学習し，異常と正常の識別を行なっている．しかし，この手法では特徴ベクトルとしてセッションごとに抽出した特徴ベクトルを使用しているため，大規模ネットワークに適用した際に特徴抽出処理に時間がかかる可能性がある．

3.　提案手法

本稿では，パケットのヘッダから単位時間ごとに抽出した特徴ベクトルを LSTM で学習し，LSTM の出力を基に算出した異常度を用いて異常を検出する手法を提案する．本手法では，パケットのヘッダから単位時間ごとに特徴を抽出している．これにより，セッションごとに特徴を抽出する手法と比べ，特徴抽出の処理にかかるコストを削減している．

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，学習に用い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から特徴ベクトルを抽出し，正規化する．その後，抽出した特徴ベクトルを LSTM により学習する．検出時の処理では，学習に用いたデータとは別の期間に収集されたトラフィックデータを学習時と同様に一定の単位時間で分割し，特徴ベクトルを抽出し，正規化する．その後，学習済みの LSTM を用いて予測値を出力する．予測値と実際の特徴ベクトルとの誤差を算出し誤差が閾値未満であった場合，その区間は正常であるとする．一方，誤差が閾値以上の場合はその区間は異常であるとして，異常の発生を検出する．

3.1　特徴ベクトルの抽出

パケットのヘッダから単位時間ごとに特徴ベクトルを抽出する．パケットから特徴ベクトルを抽出する方法として，セッションごとに特徴ベクトルを抽出する方法と単位時間ごとに特徴ベクトルを抽出する方法の大きく 2 種類がある．セッションごとに抽出した特徴ベクトルを用いる場合，大規模ネットワークに適用した際に特徴ベクトルの抽出処理に時間がかかる可能性がある．一方，単位時間ごとに特徴ベクトルを抽出する場合，多数のセッションごとに特徴ベクトルを抽出する必要がないため，少ない処理時間で特徴ベクトルを抽出することができる．また，セッションごとに特徴ベクトルを抽出する場合は UDP のようなセッションを用いない通信での異常を検出できない．そこで，本手法では単位時間ごとに特徴ベクトルの抽出を行った．特徴ベクトルの抽出の流れは，まずあるネットワークに対する異常を検知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 で収集したトラフィックデータを分割する．単位時間で分割したものを 1 区間とし，特徴ベクトル $\boldsymbol{t}$ を N 区間分収集する．各区間に含まれるパケットから表 1 に示すパケット数や宛先ポート番号種類数などの 53 種類の特徴ベクトルを抽出する．パケットのヘッダの特徴ベクトル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できる．

表 1　抽出した特徴ベクトルの一覧
Table 1　List of 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	

3.2　特徴ベクトルの正規化

学習の前に，抽出した特徴ベクトルに対して正規化を行う．正規化を行う前の特徴ベクトルは，特徴間で特徴の値の範囲が大きく異なっている．よって，正規化をせずに学習を行うと学習がうまく行われない可能性がある．そのため，各特徴ベクトルの特徴の値の範囲を一定にするために正規化を行う．ここでは，各特徴ベクトルの値の最大値が 1，最小値が 0 になるように正規化した．正規化に用いた式を式 (1) に示す．

$$f_i^{'} = \frac{f_i - min_i}{max_i - min_i} \tag{1}$$

ここで，f_i は正規化前の特徴ベクトルの i 番目の要素の値，$f_i^{'}$ は正規化後の特徴ベクトルの i 番目の要素の値，max_i，min_i はそれぞれ正規化前の特徴ベクトルの i 番目の要素の最大値，最小値を表す．

3.3　LSTM による学習

特徴ベクトルの抽出および正規化の後，RNN の拡張である LSTM による学習を行う．RNN は時系列データの処理に適したニューラルネットワークである．本手法で用いる特徴ベクトルは時間ごとに変化する時系列データであるため，RNN で学習することが適していると考えられる．一般的なニューラルネットワークでは過去の時刻の情報を保持した学習ができなかった．一方，RNN は隠れ層のノード同士に結合があるため，過去の隠れ層の出力と現在の入力の両方を参照し学習を行うことができる．しかし，従来の RNN は学習するデータが多くなると誤差逆伝播で学習する経路が長くなり，誤差が消失してしまい正しく学習できなくなる可能性がある．そのため，従来の RNN はある一定以上の過去の情報を保持できないという問題点があった．そこで本手法では，従来の RNN より長期的な情報を保持することができる LSTM を用いて学習を行った．提案手法における LSTM の学習方法を図 2 に示す．特徴ベクトル $\boldsymbol{t}_n$，$\boldsymbol{t}_{n+1}$，$\boldsymbol{t}_{n+2}$ の連続する 3 区間の特徴ベクトルを LSTM に入力し，次の区間の予測値 $\boldsymbol{y}_n$ を得る．その後，得られた予測値 $\boldsymbol{y}_n$ と入力した区間の次の区間の特徴ベクトル $\boldsymbol{t}_{n+3}$ を比較し，誤差を算出する．誤差の算出には平均二乗誤差（MSE）を用いた．平均二乗誤差の算出方法を式 (2) に示す．

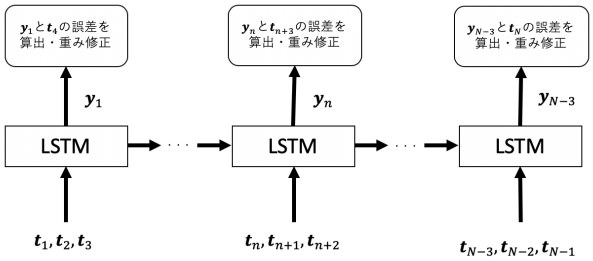

図 2　LSTM の学習の概要
Fig. 2　Overview of Learning of LSTM.

$$MSE = \frac{1}{N-3}\sum_{n=1}^{N-3}(\boldsymbol{t}_{n+3} - \boldsymbol{y}_n)^2 \tag{2}$$

そして，得られた誤差を基に誤差逆伝播により LSTM の重みの更新を行い学習を進める．学習は事前に定めたエポックに達した場合に終了することとした．LSTM ユニット内の活性化関数はシグモイド関数および tanh 関数を用いた．また，隠れ層の LSTM ユニット数は 120 ユニットとし，勾配降下法の最適化アルゴリズムには Adam を用いた．

3.4　異常検出

学習した LSTM を用いて異常を検出する．学習時の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 LSTM の学習を行ったため，検出に用いるトラフィックデータから抽出した特徴ベクトルが学習時と類似していた場合は，正しく 1 区間後の特徴ベクトルを予測することができる．一方，類似していない場合は正しく 1 区間後の特徴ベクトルを予測することができないと考えられる．そこで，予測値と 1 区間後の特徴ベクトルとの誤差を異常度とし，異常度が閾値以上であった場合を異常として検出する．異常検出の例を図 3 に示す．図 3 の縦軸が異常度，横軸が区間数，また点線は閾値を表している．このように，設定した異常

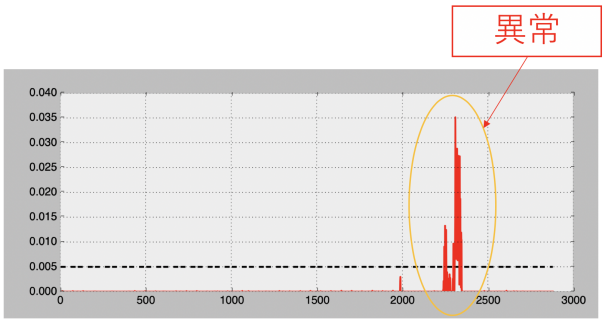

図 3　異常検出の例
Fig. 3　Example of Anomaly Detection.

度が閾値を上回っている区間を異常，下回っている区間を正常として識別する．異常度を算出する式を式 (3) に示す．

$$Anomaly\ score = \frac{1}{2}(\boldsymbol{t}'_{n+3} - \boldsymbol{y}'_n)^2 \qquad (3)$$

ここで，$\boldsymbol{t}'_{n+3}$ は入力された特徴ベクトルの 1 区間後の特徴ベクトル，$\boldsymbol{y}'_n$ は学習済み LSTM に特徴ベクトル $\boldsymbol{t}'_n$，$\boldsymbol{t}'_{n+1}$，$\boldsymbol{t}'_{n+2}$ を入力した際の予測値である．

4.　実験

本手法の有効性を確認するために MWS 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．

4.1　MWS データセットにおける実験

MWS データセットの BOS データセット [12] を使用した．BOS データセットは，マルウェアを実行したホストの通信をキャプチャしたものである．また，BOS データセット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．進行度ごとの説明を表 2 に示す．本稿では，進行度 2 のトラフィックデータを異常を含まない正常なトラフィックデータ，進行度 7，8 のトラフィックデータを異常を含むトラフィックデータとして実験に用いた．各進行度で 24 時間のトラフィックデータを使用した．また，特徴ベクトルの抽出の際の単位時間は 30 秒とし 2880 区間に分割し実験に用いた．本研究では，C&C サーバとの通信を含む区間を異常，それ以外の区間を正常として実験した．

評価方法は ROC（Receiver Operating Characteristic）曲線および AUC（Area Under the Curve）と Precision，Recall，F-measure を用いた．ROC 曲線とは正常と異常を分類する際の閾値を変化させたときの TPR（True Positive Rate）と FPR（False Positive Rate）を計算し，プロットしたグラフである．ここで，TPR とは異常データに対して正しく異常と識別した割合であり，FPR とは正常データを誤って異常と識別した割合を示す．ROC 曲線は左上に近づくほど検出精度が高いことを示している．また ROC 曲線が左上に近づいていることを示す指標として AUC 値を用いる．AUC 値は ROC 曲線の下側の面積を算出した値であり，この値が 1 に近いほど完全に分類できていることを示す．また，実験では 3.4 節で説明した異常検出の際の閾値を変化させ TPR と FPR を算出し，ROC 曲線を描画した．Precision は異常であると識別されたものの中で，異常であったものの割合を表す．Precision の算出式を式 (4) に示す．

表 2　進行度の説明
Table 2　Explanation of Progress.

進行度	説明
1，2	通信発生なし
3，4，5	通信発生したが，C&C サーバとの攻撃通信不成立
6，7，8	通信発生かつ C&C サーバとの攻撃通信成立

$$Precision = \frac{TP}{TP + FP} \qquad (4)$$

Recall は異常を正しく異常であると識別できた割合を表す．Recall の算出式を式 (5) に示す．

$$Recall = \frac{TP}{TP + FN} \qquad (5)$$

ここで，TP（True Positive）は異常を正しく異常と識別した数，FP（False Negative）は異常と識別したが正常であった数，TN（True Negative）は正常を正しく正常と識別した数，FN（False Negative）は正常と識別したが異常であった数を表す．

F-measure は Precision と Recall の調和平均である．F-measure の算出式を式 (6) に示す．

$$F - measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \qquad (6)$$

4.1.1　実験結果

進行度 2 のトラフィックデータを用いて学習した際のエポック数に対する学習誤差の遷移を図 4 に示す．エポック数が 30 を超えた辺りから誤差の遷移が緩やかになり徐々に収束していることを確認できた．MWS データセットを用いた実験において，エポック数は 30 で実験を行った．

学習データとして進行度 2 のトラフィックデータ，テストデータとして進行度 8 のトラフィックデータを用いて ROC 曲線および AUC 値の算出を行った．テストデータのラベル付けとして 1 区間に C&C サーバとの通信のパケット数が P 以上の場合に異常とした．P の値を 1，10，20 と変化させたときの ROC 曲線および AUC 値を図 5 に示す．図 5 より，1 区間に 10 パケット以上の C&C サーバとの通信が含まれるときを異常とした際に AUC 値が 0.93 と最も高くなった．また，1 区間に 20 パケット以上の場合は AUC 値が 0.88 となり，10 パケット以上の場合より減

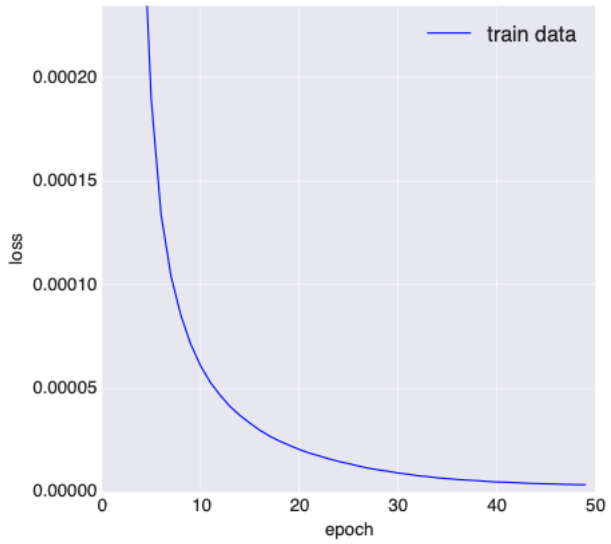

図 4　学習誤差の遷移（MWS2018 Dataset）
Fig. 4　Transition of Training Error(MWS2018 Dataset).

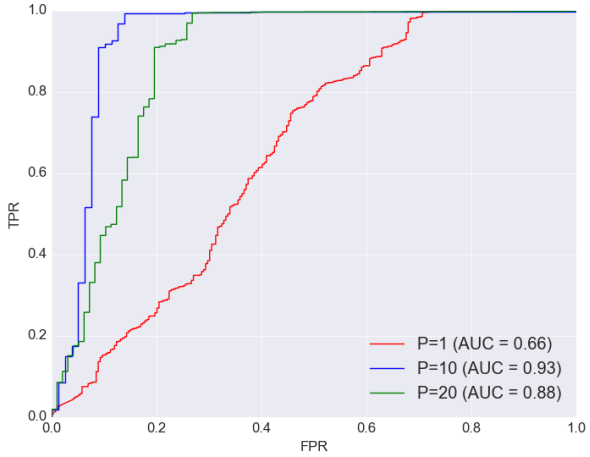

図 5　P の値ごとの ROC 曲線
Fig. 5　ROC Curve when P is changed.

少する結果となったが比較的高い AUC 値となった．ここで，1 区間に含まれている総パケット数は区間により異なるため，1 区間の総パケット数に対する異常パケットの割合についても確認した．1 区間に 10 パケットの C&C サーバとの通信が含まれる区間の総パケット数に対する異常パケットの割合の最小値は 8%であった．また，20 パケットの C&C サーバとの通信が含まれる区間の総パケット数に対する異常パケットの割合の最小値は 10%であった．これらより，C&C サーバとの通信のパケットが 1 区間にある程度以上含まれていた場合，正常な区間から抽出した特徴ベクトルとの差異が生まれ，高い精度で分類できることを確認できた．一方で，1 区間に C&C サーバとの通信のパケットが 1 パケット以上の場合は AUC 値が 0.66 と最も低かった．また，1 パケットの C&C サーバとの通信が含まれる区間の総パケット数に対する異常パケットの割合の最小値は 0.2%であった．これは，1 区間に含まれる C&C サーバとの通信のパケットが少なく，抽出した特徴ベクトルが正常の特徴ベクトルと類似したためであると考えられる．

学習データとテストデータの組み合わせごとの AUC，Precision，Recall，F-measure を表 3 に示す．また，テストデータに対するラベル付けは 1 区間に C&C サーバとの通信のパケット数が 10 パケット以上含まれる場合を異常として検出実験を行った．異常検出の際の閾値は図 5 の P=10 のグラフにおいて TPR が 0.91，FPR が 0.09 の際の閾値である 0.00026 を全てのデータの組み合わせで用いた．進行度 2 のトラフィックデータで学習を行い，進行度 7 および進行度 8 のトラフィックデータでテストを行った結果，いずれの値も 0.80 以上と高い結果となった．また，進行度 7 のトラフィックデータで学習を行い，進行度 8 のトラフィックデータでテストを行った結果，いずれの値も低い結果となった．また，進行度 8 のトラフィックデータで学習を行い，進行度 7 のトラフィックデータでテストを行った結果，AUC，Precision の値については高いという結果になった．また，Recall および F-measure の値についても AUC，Precision には劣るが比較的高い結果となった．

進行度 2 のトラフィックデータで学習を行い，進行度 7 および進行度 8 のトラフィックデータでテストを行った際の結果について考察する．いずれの値も高い結果となったのは学習データが全て正常な通信であったからであると考えられる．そのため，正常な通信の特徴をうまく学習することができ，テストデータに現れた異常を正しく識別することができた．次に，進行度 7 のトラフィックデータで学習を行い，進行度 8 のトラフィックデータでテストを行った結果について考察する．いずれの値も低い結果となったのは，学習に用いた進行度 7 のトラフィックデータの中に多くの異常な通信が含まれていたからであると考えられる．ここで，進行度 7 および進行度 8 のトラフィックデータにおけるラベルの内訳を表 4 に示す．表 4 より，進行度 7 のトラフィックデータは 2880 区間の内，1650 区間に正常のラベル，1230 区間に異常のラベルが付与されている．そのため，正常の特徴と異常の特徴の両方を学習してしまい，うまく検出することができなかった．進行度 8 のトラフィックデータで学習を行い，進行度 7 のトラフィックデータでテストを行った結果について考察する．AUC，Precision の値が高くなったのは正常の特徴をうまく学習できていたためであると考えられる．表 4 より，学習に用いた進行度 8 のトラフィックデータの 2880 区間の内，2801 区間に正常のラベル，79 区間に異常のラベルが付与されており，ほとんどが正常な区間であることがわかる．そのため，正常な特徴が学習されてうまく検出することができた．一方で，Recall および F-measure の値がそれぞれ 0.62，0.76 と比較的低い値となったのは，学習データとして進行度 2 のトラフィックデータ，テストデータとして進

表 3　検出結果
Table 3　Result of Detection.

学習データ	テストデータ	AUC	Precision	Recall	F-measure
進行度 2	進行度 7	0.97	0.99	0.85	0.92
進行度 2	進行度 8	0.93	0.80	0.84	0.81
進行度 7	進行度 8	0.73	0.13	0.58	0.21
進行度 8	進行度 7	0.91	0.99	0.62	0.76

表 4　ラベルごとの区間数
Table 4　Number of Sections per Label.

データ	正常ラベル	異常ラベル	合計
進行度 7	1650 区間	1230 区間	2880 区間
進行度 8	2801 区間	79 区間	2880 区間

行度 8 のトラフィックデータを用いた際に最適であった閾値を用いたためであると考えられる．そのため，閾値の調整により改善できると考えられる．以上の結果より，学習に用いるトラフィックデータに異常が含まれていない，もしくは異常が少数であれば異常を検出可能であることを確認できた．一方で，学習に用いるトラフィックデータに異常が多数含まれてる場合はうまく検出できなかった．

4.2　大阪府立大学のトラフィックデータにおける実験

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収集したトラフィックデータを使用して実験を行った．学習データとして 2019 年 7 月 8 日 7 時 25 分 1 秒から収集した 10 分のトラフィックデータを用いた．テストデータとしては 2019 年 7 月 8 日 7 時 35 分 1 秒から収集した 10 分のトラフィックデータを用いた．また．テストデータ内の 7 時 41 分 46 秒に UTM（Unified Threat Management）で DoS 攻撃の一種である TCP flood 攻撃が観測されている．学習データ，テストデータ共に特徴ベクトルの抽出の際の単位時間は 0.5 秒とし 1200 区間に分割して実験に用いた．UTM において DoS 攻撃を観測した区間はテストデータの 810 区間目である．また，実運用中のトラフィックを学習およびテストに用いているため正解ラベルを付与することができない．そのため，AUC 値や Recall 等の評価指標を使用することができない．そこで，トラフィックデータの異常度を算出し，どのような特徴があるのかを確認した．

4.2.1　実験結果

学習データをエポック数を 300 として学習した．学習したモデルにテストデータを入力した際の異常度のグラフを図 6 に示す．図 6 を見ると，いくつかの区間で異常度が高くなっていることがわかる．その中でも，800 区間の直前から 800 区間にかけて異常度の値が非常に大きくなっていた．これは，UTM が観測した DoS 攻撃の影響であると考えられる．UTM がアラートを発生したのは 810 区間目であるが，UTM における検出はある一定の閾値を超えた場合のみに行われる．そのため，実際に DoS 攻撃が発生した時刻はアラート発生よりも前の時刻であるため，800 区間の直前から 800 区間にかけて異常度の値が高くなったと考えられる．また，その他にも異常度が高い区間がいくつか存在した．これは，本手法により UTM では検知できなかった異常を検出できた可能性があると考えられる．今後，用いたトラフィックデータや実験結果を精査していく必要がある．

5.　まとめ

本稿では，パケットのヘッダから抽出した特徴ベクトルに対して LSTM で学習を行うことで異常を検出する手法を提案した．実験では，MWS データセットおよび大阪府立大学のトラフィックデータの 2 種類のデータを用いて本手法の有効性を確認した．MWS データセットを用いた実験では，1 区間に含まれている異常な通信が一定以上あれば高い AUC 値で検出できることを確認した．さらに，学習に用いるトラフィックデータに異常が含まれていない，もしくは異常が少数であれば異常を検出可能であることを確認できた．また，大阪府立大学のトラフィックデータを用いた実験では，UTM が観測した異常を検出することができることを確認した．

今後の課題としては学習の際のパラメータや閾値の調整や実運用中のトラフィックに適用した際の実験結果の精査などが挙げられる．

参考文献

[1] Snort, <https://www.snort.org/>(参照 2019-07-15).

[2] Suricata, <http://suricata-ids.org/>(参照 2019-07-15).

[3] 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).

[4] 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).

[5] 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).

[6] 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).

[7] Mitsuhiro Hatada, 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).

[8] 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).

[9] 谷澤俊樹, 青木茂樹, 宮本貴朗：パケットのヘッダ情報に

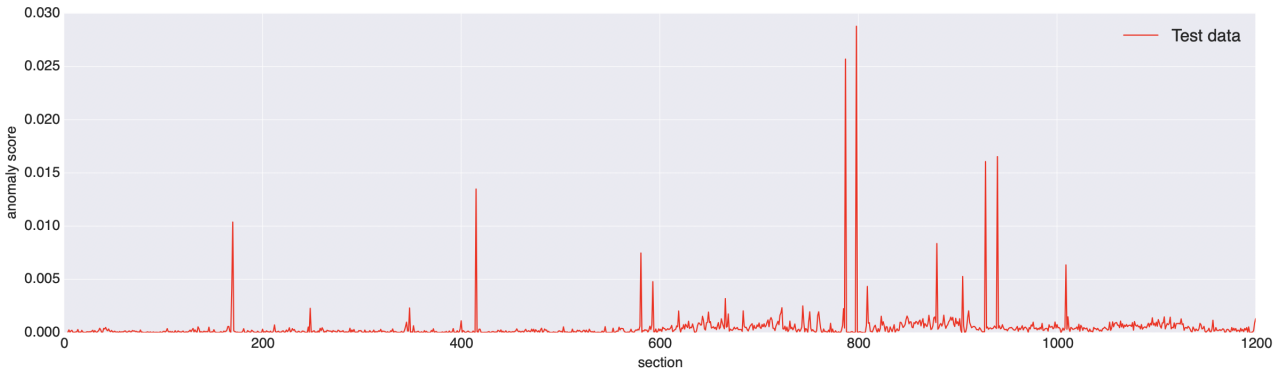

図 6　異常度のグラフ（大阪府立大学のトラフィックデータ）

Fig. 6　Graph of Anomaly Score (Traffic Data of Osaka Prefecture University).

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).

[10] Hongyu Liu, Bo Lang, Ming Liu, Hanbing Yan:CNN and RNN based payload classification methods for attack detection, Knowledge-Based Systems Volume 163, 1 January 2019, 332-341(2019).

[11] C. Xu, J. Shen, X. Du, F. Zhang:An intrusion detection system using a deep neural network with gated recurrent units, IEEE Access, vol.6, pp.48697-48707, (2018).

[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).