LSTMを用いた不審なWebアクセスの検出

中前 諒哉[1,a)]　青木 茂樹[1,b)]　宮本 貴朗[1]

概要：近年，標的型攻撃による被害が深刻化している．標的型攻撃はシグネチャベースによる事前対策だけでは検出が難しい．その為，マルウェアに感染した後の端末の早期発見が重要視されている．マルウェア感染後にはC2サーバへのアクセスなど端末が感染前と異なる通信を行うケースが多い．そこで本稿では，感染前の端末のWebアクセス履歴を学習し，感染後の端末の不審なWebアクセスを検出する手法を提案する．まず，Proxyサーバのログを端末ごとに分類し，端末ごとに感染前のWebアクセス履歴を作成する． Webアクセス履歴から，ドメインなどの7種類の単語によりコーパスを作成する．作成したコーパスをWord2vecで学習して単語ベクトルとし，単語ベクトルを基に算出する値や受信バイト数などの10種類の特徴量を抽出する．次に，抽出した特徴量の時系列データをリファレンスデータとしてLSTMで学習する．検出対象のWebアクセスログに対しても同様に特徴量を抽出し，リファレンスデータから算出した予測値との差がしきい値以上の場合を不審なWebアクセスとして検出する．実験では，本学の職員と学生のWebアクセス履歴を用いて，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，深層学習，LSTM, Word2vec

Suspicious web access detection using LSTM

Ryoya Nakamae[1,a)]　Shigeki Aoki[1,b)]　Takao Miyamoto[1]

Abstract: APT(Advanced Persistent Threat) attacks have become serious problem in the world. It is difficult to detect them by signature-based proactive measures. Thus, it is important to detect infected terminals after malware infection as soon as possible. Web accesses of infected terminals are different from accesses of uninfected terminals, since C2 communications are increasing. First, we extract feature vectors from sequence of web accesses of each terminal by method based on Word2Vec. Next, we train LSTM with the extracted feature vectors as reference data. After that, we compare feature vectors extracted from new sequence of web accesses with predicted value of the LSTM. We detect suspicious accesses when the difference is greater than threshold value. In the experiment, we confirmed the effectiveness of our method using sequence of web accesses of faculty members and students of Osaka Prefecture University.

Keywords: Anomaly Detection of Network, Deep Learning, LSTM, Word2vec

1. はじめに

マルウェアによるサイバー攻撃が年々深刻化しており，サイバー攻撃に対する社会の関心が高まっている．従来のサイバー攻撃は，不特定多数のホストの中からサイバー攻撃に対する対策が不十分なホストを探索して攻撃を行っていた．そのため多くの組織は，組織内ネットワークと組織外ネットワークの境界にFirewallやIDS（Intrusion Detection System）などの侵入を妨害・検知するシステムを設置し，ホストにはウイルス対策ソフトを稼働させることで，組織内ネットワークのセキュリティを確保してきた．しかしながら，近年問題となっているサイバー攻撃の一種

[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)] saa01183@edu.osakafu-u.ac.jp
[b)] aoki@kis.osakafu-u.ac.jp

である標的型攻撃は，従来の対策だけでは検出が難しい．標的型攻撃は特定の組織を狙った攻撃であり，攻撃者は遠隔操作型マルウェアを組織内ネットワークに送り込む．そして，送り込んだマルウェアを組織の人物に実行させることによって，端末をマルウェアに感染させる．組織内の端末をマルウェアに感染させると，攻撃者はその感染端末を踏み台にして，組織内ネットワークに存在する窃取したい情報を探し出し，情報を窃取する．

マルウェアへの感染前と感染後の端末の通信は変化すると考えられる．感染後の変化に迅速に気付き，被害を最小限に抑える為には，攻撃者による組織内ネットワークへの侵入後の不審な通信を速やかに検出する必要がある．一般に，標的型攻撃に用いられるマルウェアは C2 サーバと HTTP による通信を行っている事が多いため，マルウェアに感染後の不審な通信は Web アクセスを解析することにより検出できると考えられる．

文献 [1] では Proxy サーバのログから Web アクセスに関する特徴量を抽出し，標的型攻撃を検出する手法を提案している．この手法ではログから特徴量を抽出する際に，one-hot encoding を用いているため学習に用いる特徴量の次元が膨大となっている．

文献 [2] では Proxy サーバのログから Web アクセスに関する特徴量を Doc2vec によって抽出し，標的型攻撃を検出する手法を提案している．この手法は，C2 通信の検知に優れているものの，攻撃の種類が増加すると検知率が低下することが課題として挙げられている．

文献 [3] では，CNN（Convolutional Neural Network）と RNN（Recurrent Neural Network）の拡張である LSTM（Long short-term memory）を用いてネットワークの異常を高精度に検出する手法を提案している．

本稿では，特定の端末の Web アクセス履歴中の不審な Web アクセスを，自然言語処理技術の一つである Word2vec と，深層学習手法の一つである LSTM を用いて検出する手法を提案する．本手法では，Web アクセスの特徴を Word2vec を利用して抽出することによって特徴ベクトルの次元を低減し，Web アクセスの特徴ベクトルを LSTM で学習する事により，高精度に異常を検出する事ができる．実験では，本学の職員と学生の Web アクセス履歴を用いて，本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，マルウェア感染後の端末の不審な動きを検出するシステムの代表的な手法である文献 [1], [4] と自然言語処理技術を用いた異常検出手法の文献 [2], [5], [6] 及び深層学習を用いた異常検知手法の文献 [3], [7] について述べる．

文献 [1] では Proxy サーバのログから Web アクセスに関する特徴量を抽出し，次元圧縮を行い，次元圧縮前と次元圧縮後の行列を比較して，誤差を算出する事で標的型攻撃を検出する手法を提案している．この手法ではログから特徴量を抽出する際に one-hot encoding を用いているため，学習に用いる特徴量が 11,969 次元と膨大になっている．次元の増大に伴い，異常検知に多大な時間が必要な為，異常検知までの時間の短縮が課題となっている．

文献 [4] では，Firewall ログを解析する事により悪性リストを作成し，作成した悪性リストに一致するログが一定回数以上出力された場合に不審な動きを検出する手法を提案している．この手法では Firewall ログの動的解析結果を基に悪性リストを作成しているため，検知を回避するようにカスタマイズされている RAT（Remote Administration Tool）や未知のマルウェアには対応できない事が課題となっていた．

文献 [1] で問題となっている，検出精度の低下や処理時間の増大に対応するために，次元数の増大を低減できる，Doc2vec[2]，IP2Vec[5]，LSI[6] などの自然言語処理技術を応用した手法が提案されている．

文献 [2] では，Proxy サーバの正常な通信ログと不正な通信ログから，Doc2Vec によって特徴ベクトルを抽出し，抽出したベクトルをサポートベクタマシン（SVM）やランダムフォレスト（RF）等のモデルで学習する手法を提案している．そして，検知対象のログから抽出した特徴ベクトルを学習モデルに入力して，異常を検出している．この手法は C2 通信の特徴の検知に優れているものの，攻撃の種類が増加すると検知率が低下することが課題として挙げられている．

文献 [5] では，セキュリティ検知機器のアラートログを自然言語とみなして Word2vec を応用した手法を適用してアノマリを検知する手法を提案している．IP アドレスごとのアラート情報を Word2vec を基に改良した IP2vec に入力することで IP アドレスごとに特徴ベクトルを作成し，ベクトル空間上の重心からの距離をアノマリスコアとして算出している．この手法は，IP アドレスごとの異常を検知できるものの，端末の動きの変化の検出に適用することは難しい．

文献 [6] では，標的型メールの添付ファイルとして利用されることが多い，MS 文書の悪性マクロを検出する手法を提案している．この手法では既知の良性マクロの特徴を LSI(Latent Semantic Indexing) を用いてベクトル化し，そのベクトルを One-class SVM や Local Outlier Factor（LOF）で学習する．そして，検査に用いる MS 文書から抽出したベクトルを学習後の One-class SVM または LOF に入力し，外れ値を異常として検出している．

文献 [3] では，パケットのペイロードから抽出した特徴に対して CNN を適用した手法と RNN の拡張である LSTM を適用した手法を提案し，ネットワークの異常を検出している．文献 [7] では，RNN の拡張である GRU（Gated

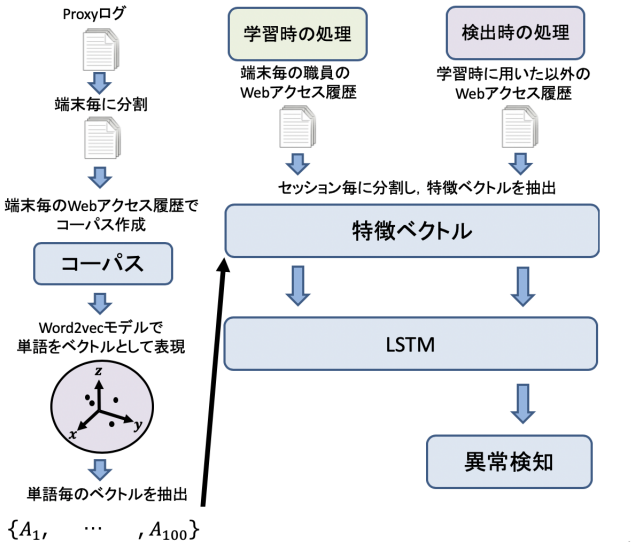

図 1　提案手法の概要
Fig. 1　Outline of Proposal Method.

Recurrent Unit）と多層パーセプトロンを組み合わせたネットワーク異常検知手法を提案している．これらの手法はネットワークの異常を深層学習手法により高精度に検出できている．これらの深層学習手法を Web アクセスの特徴解析に応用することにより，端末の不審な動きを高精度に検出できると考えられる．

3. 提案手法

本研究は，組織内に設置されている端末の Web アクセス履歴から，マルウェア感染の疑いのある端末を検出することを目的としている．個人が利用する端末の Web アクセス履歴には個人ごとに異なる特徴が現れると考えられる．そして，マルウェア等に感染した場合には端末を使用している人物の特徴とは異なる Web アクセス履歴が観測されると考えられる．そこで本稿では，特定の端末の Web アクセス履歴を学習して普段の挙動とは異なる不審な Web アクセスを検出することによって，マルウェアの感染を検出する手法を提案する．提案手法の概要を**図 1** に示す．本手法では対象端末の Web アクセス履歴からドメイン，クエリ，パスなどの 7 種類の単語によりコーパスを作成する．作成したコーパスを自然言語処理技術の一つである Word2vec で学習することにより，Web アクセス履歴の単語を単語ベクトルとして表現する．次に，Web アクセス履歴をセッションごとに分割し，単語ベクトルを基に算出する値や受信バイト数などの 10 種類の特徴量を抽出する．抽出した特徴量の時系列データをリファレンスデータとし，深層学習の一つである LSTM で学習する．検出対象の Web アクセスログに対しても同様に特徴量を抽出し，リファレンスデータから算出した予測値との差がしきい値以上の場合に不審な Web アクセスとして検出する．

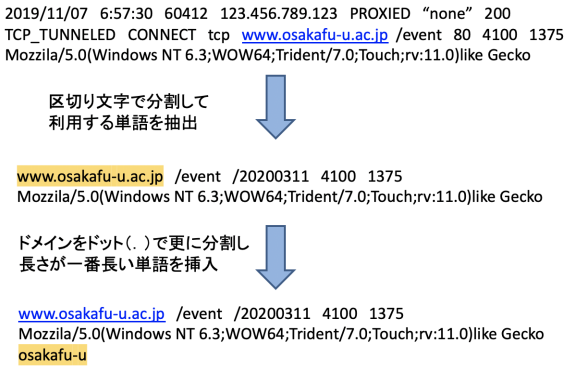

図 2　Web アクセス履歴の単語抽出の概要
Fig. 2　Overview of word extraction.

3.1 Web アクセス履歴からのコーパスの作成

Web アクセス履歴からコーパスを作成する手順について述べる．本研究では，Proxy サーバのログを端末ごとに分割した通信履歴を Web アクセス履歴として利用する．まず，Proxy ログから対象端末のログを抽出し，対象端末の Web アクセス履歴を作成する．次に，対象端末の Web アクセス履歴からコーパスを構成する単語を抽出する．単語抽出の流れを**図 2** に示す．Web アクセス履歴内に記載されている文章を区切り文字で分割し，本手法で利用する単語のフィールドを抽出する．Web アクセス履歴の中から抽出する単語の種類は，ドメイン，クエリ，パス，受信バイト，送信バイト，UserAgent，ドメインの中で一番長い単語の計 7 種類とする．ドメインの中で一番長い単語を利用する理由は企業名等が多く含まれ，ドメインの特徴を大きく表していると考えられる為である．以上の処理を，対象端末の全 Web アクセス履歴に対して行い，端末ごとにコーパスを作成する．

3.2 単語ベクトルの抽出

前節で作成した端末ごとのコーパスを Word2Vec で学習することで，端末ごとに単語ベクトル空間を作成する．Word2vec は，隠れ層が 1 層，出力層が 1 層のニューラルネットワークのモデルであり，文脈を考慮して単語を学習し，単語をベクトルとして表現できる．Word2Vec のベクトル空間では同じ使われ方をする単語は類似したベクトルになる．本手法では，文献 [5] の手法と同様に Word2vec の学習アルゴリズムの 1 つである Skip-Gram モデルでコーパスを学習する．Skip-Gram モデルを**図 3** に示す．例として，p 個の単語を含む文章が与えられているとする．各単語を 1 対 1 に自然数へ対応させることで，単語を $w_i(\in\{1,\cdots,p\})$ と表現する．w_i に対応する one-hot ベクトルを $\boldsymbol{x}_i \in \{0,1\}^p$ とする．$\boldsymbol{x}_i$ は i 番目の要素が 1，それ以外の要素が 0 の p 次元ベクトルである．ネットワークへの入力層 $\boldsymbol{x} \in \mathbb{R}^p$

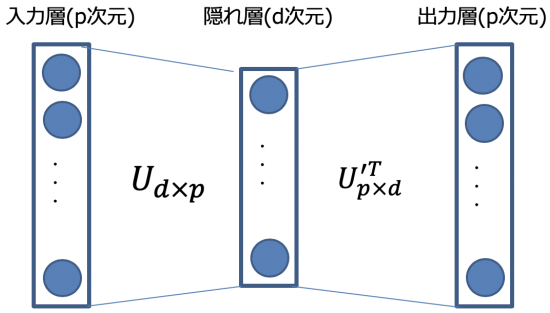

図 3　Skip Gram モデル
Fig. 3　Skip-Gram Model.

には，単語の one-hot ベクトルが入力される．隠れ層，出力層の出力 $\boldsymbol{h}$，$\boldsymbol{y}$ は，重み行列 $\boldsymbol{U} = (\boldsymbol{u}_1,\cdots,\boldsymbol{u}_p) \in \mathbb{R}^{d\times p}$，$\boldsymbol{U}' = (\boldsymbol{u}'_1,\cdots,\boldsymbol{u}'_p) \in \mathbb{R}^{d\times p}$ を用いてそれぞれ，

$$\boldsymbol{h} = \boldsymbol{U}\boldsymbol{x} \in \mathbb{R}^d \tag{1}$$

$$\boldsymbol{y} = \mathrm{softmax}\left(\boldsymbol{U}'^{T}\boldsymbol{h}\right) = \frac{\exp\left(\boldsymbol{U}'^{T}\boldsymbol{h}\right)}{\sum_k \exp\left(\boldsymbol{u}'^{T}_k\boldsymbol{h}\right)} \in \mathbb{R}^p \tag{2}$$

と表現する．各単語 w_i に対する $\boldsymbol{h}_i$ が，ベクトル表現された単語の特徴量となる．$\boldsymbol{h}_i$ を以下では，単語ベクトルと呼ぶ．

skip-gram モデルでのネットワークの学習は，単語が与えられた時，その単語の前後に周辺語が共起する尤度に基づいて損失関数を定義し，損失関数の最小化によってパラメータ（$\boldsymbol{U}$，$\boldsymbol{U}'$）を学習する．単語 w_i と，周辺語が共起する確率 I は，以下の式で与えられる．

$$I = \prod_{\boldsymbol{w}_{c,i}\in C(w_i)} P(w_{c,i}|w_i) \tag{3}$$

$$P(w_{c,i}|w_i) = \frac{\exp\left(\boldsymbol{u}'^{T}_{c,i}\boldsymbol{u}_i\right)}{\sum_k \exp\left(\boldsymbol{u}'^{T}_k\boldsymbol{u}_i\right)} \tag{4}$$

ここで，$\boldsymbol{u}'_{c,i} := \boldsymbol{U}'_{w_{c,i}}$，即ち行列 $\boldsymbol{U}'$ の第 $w_{c,i}$ 列と表記する．$P(w_{c,i}|w_i)$ は，ネットワークに $\boldsymbol{x}_i$ を入力したときの，出力層 $\boldsymbol{y}$ における $w_{c,i}$ 番目のノードに対応する．Word2vec の最適化は，以下の負の対数尤度 L_{w2v} の最小化によって実現する．

$$L_{w2v} = -\sum_i \sum_{w_{c,i}\in C(w_i)} \log P(w_{c,i}|w_i) \tag{5}$$

最適化の結果，最適な埋め込み行列 $\bar{\boldsymbol{U}} = (\bar{\boldsymbol{u}}_1,\cdots,\bar{\boldsymbol{u}}_p)$ が得られる．$\bar{\boldsymbol{U}}$ を用いて，単語 w_i の特徴量を $\boldsymbol{h}_i = \bar{\boldsymbol{U}}\boldsymbol{x}_i = \bar{\boldsymbol{u}}_i$ と表現する．

以上の処理で抽出した単語ベクトルの空間では類似する使われ方をする単語の距離は短く，類似しない使われ方の単語の距離は長くなる．したがって，アクセスの傾向が類似する Web ページ間の距離は短く，アクセスの傾向が異なる Web ページ間の距離は長くなる．

表 1　抽出した特徴量の一覧
Table 1　List of Features.

特徴量
セッション中の受信データサイズの最大値
セッション中の受信データサイズの最小値
セッション中の送信データサイズの最大値
セッション中の送信データサイズの最小値
セッション中のリクエストタイムの最大値
セッション中のリクエストタイムの平均値
セッション内の POST メソッドでのアクセスの割合
セッション内の GET メソッドでのアクセスの割合
セッション内の CONNECT メソッドでのアクセスの割合
セッション内の各ログ中のドメインが遷移する様子を単語ベクトル空間での距離で表現した値の最大値

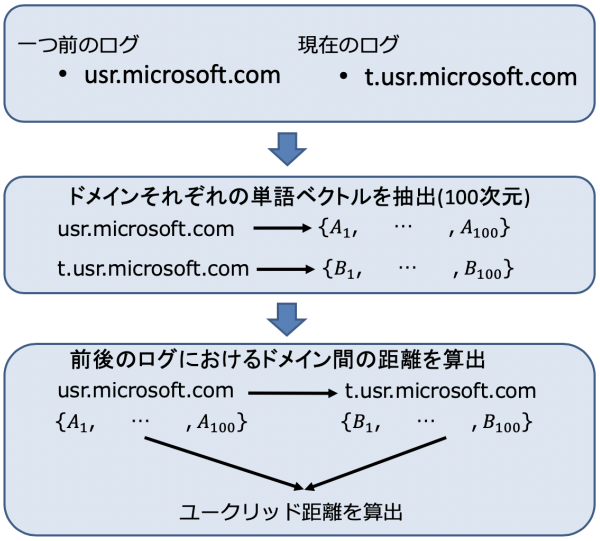

図 4　単語ベクトルの距離の算出例
Fig. 4　Example of Calculating distance between word vectors.

3.3　セッション分割及び特徴量抽出

Web アクセス履歴から LSTM に学習させるための特徴量を抽出する．本研究では，文献 [1] と同様にセッション単位で特徴量を抽出する．そして，同一 IP アドレスからの通信が 30 秒空いた場合，別のセッションとして扱うことによりセッションを抽出する．端末のログをセッションごとに分割し，n 番目のセッションから表 1 に示す 10 次元の特徴量 $\boldsymbol{t}_n$ を抽出する．表 1 中の 10 行目の特徴量は，セッション内の各ログから抽出したドメイン名の遷移に関する値である．セッション内でのドメインの遷移をユークリッド距離として算出する例を図 4 に示す．図 4 では例として usr.microsoft.com から t.usr.microsoft.com に遷移した時の算出方法を説明している．学習が終了した Word2vec のベクトル空間から usr.microsoft.com の単語ベクトル $\boldsymbol{A}$(100 次元) と，t.usr.microsoft.com の単語ベクトル $\boldsymbol{B}$(100 次元) を抽出する．二つのベクトルからユーク

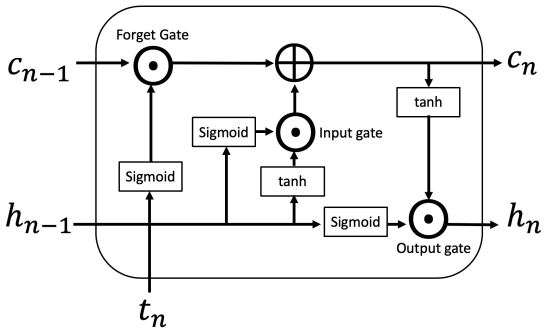

図 5　LSTM の内部構造
Fig. 5　Internal structure of LSTM.

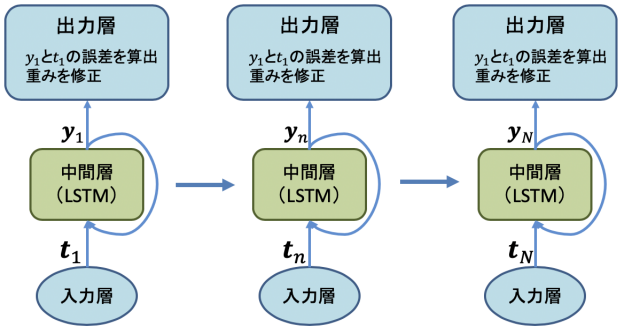

図 6　LSTM の学習の概要
Fig. 6　Outline of Training LSTM.

リッド距離を算出する．この処理をセッション内のドメインの遷移全てに対して行い，セッション内でのドメインの遷移の最大値を特徴量の一つとして用いる．

3.4　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を各特徴ベクトルの値の最大値が 1，最小値が 0 になるように次式で正規化する．

$$f_i' = \frac{f_i - min_i}{max_i - min_i} \tag{6}$$

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．

3.5　LSTM による Web アクセス履歴の学習

LSTM[8] によりマルウェア感染前の Web アクセス履歴を学習する．LSTM とはネットワーク内部での短期記憶を長期間活用できる構造を持つ RNN の一種である．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量はセッションごとに変化する時系列データであるため，RNN，LSTM などの時系列データを扱える深層学習手法での学習が適していると考えられる．RNN と LSTM の大きな違いは，記憶セル（memory cell）を導入し，中間層の状態を長期間伝播できるようにしていることである．LSTM の記憶セルは入力ゲート（input gate），出力ゲート（output gate），忘却ゲート（forget gate）をもち，それぞれ，入力から記憶セルに書き込む量，記憶セルから出力する量，直前の時刻の記憶セルの内容を保持する量を調整している．LSTM の内部構造を図 5 に示す．セッション n の特徴ベクトル $\boldsymbol{t}_n$ に対して，入力ゲート $\boldsymbol{i}_n$，忘却ゲート $\boldsymbol{f}_n$，記憶セル $\boldsymbol{c}_n$，出力ゲート $\boldsymbol{o}_n$，中間層 $\boldsymbol{h}_n$ を次式で計算する．

$$\boldsymbol{i}_n = \sigma(\boldsymbol{W}_{it}\boldsymbol{t}_n + \boldsymbol{W}_{ih}\boldsymbol{h}_{n-1} + \boldsymbol{b}_i) \tag{7}$$

$$\boldsymbol{f}_n = \sigma(\boldsymbol{W}_{ft}\boldsymbol{t}_n + \boldsymbol{W}_{fh}\boldsymbol{h}_{n-1} + \boldsymbol{b}_f) \tag{8}$$

$$\boldsymbol{o}_n = \sigma(\boldsymbol{W}_{ot}\boldsymbol{t}_n + \boldsymbol{W}_{oh}\boldsymbol{h}_{n-1} + \boldsymbol{b}_o) \tag{9}$$

$$\boldsymbol{c}_n = \boldsymbol{f}_n \odot \boldsymbol{c}_{n-1} + \boldsymbol{i}_n \odot \tanh(\boldsymbol{W}_{ct}\boldsymbol{t}_n + \boldsymbol{W}_{ch}\boldsymbol{h}_{n-1} + \boldsymbol{b}_c) \tag{10}$$

$$\boldsymbol{h}_n = \boldsymbol{o}_n \odot \tanh(\boldsymbol{c}_n) \tag{11}$$

ここで，$\boldsymbol{W}_{it}$，$\boldsymbol{W}_{ih}$，$\boldsymbol{W}_{ft}$，$\boldsymbol{W}_{fh}$，$\boldsymbol{W}_{ot}$，$\boldsymbol{W}_{oh}$，$\boldsymbol{W}_{ct}$，$\boldsymbol{W}_{ch}$ は学習する重み行列，$\boldsymbol{b}_i$，$\boldsymbol{b}_f$，$\boldsymbol{b}_o$，$\boldsymbol{b}_c$ は学習するバイアスベクトル，σ はシグモイド関数，tanh は hyperbolic tangent 関数，$\odot$ はアダマール関数である．

図 6 は今回の研究に適用する LSTM モデルの模式図である．大きく分けて，入力層，中間層，出力層で構成されている．Web アクセス履歴の特徴ベクトル $\boldsymbol{t}_n$ をリファレンスデータ として入力層に入力し，入力値に対する予測値 $\boldsymbol{y}_n$ を得る．得られた予測値 $\boldsymbol{y}_n$ とリファレンスデータ $\boldsymbol{t}_n$ を比較し，次式で示す平均二乗誤差（MSE）を算出する．

$$MSE = \frac{1}{n-1}\sum_{i=1}^{n-1}(\boldsymbol{t}_i - \boldsymbol{y}_i)^2 \tag{12}$$

得られた誤差を基に誤差逆伝搬を行い，LSTM の重みを更新する．そして，事前に定めたエポック数に到達した時，学習を終了する．ここで，勾配降下法の最適化アルゴリズムには Adam を用いた．

3.6　異常ログの検出

3.5 節で学習した LSTM を用いて異常を検出する．検出に用いる Web アクセス履歴の特徴ベクトル $\boldsymbol{t}_n$ が LSTM の学習に用いたリファレンスデータと類似している場合は，正しい $\boldsymbol{y}_n$ を予測する事ができる．一方，類似していない場合は正しい $\boldsymbol{y}_n$ を予測できないと考えられる．3.5 節と同様に，式（12）を用いて $\boldsymbol{t}_n$ と $\boldsymbol{y}_n$ から誤差を算出し，異常度とする．異常度がしきい値 θ を上回っている時に不審な Web アクセスの発生を検知し，下回っている時を正常とする．しきい値 θ の決定には学習を行った同一の端末から抽出した正常な特徴量を用いる．正常な特徴量の異常度を算出し，算出した値の平均値 $+3\sigma$ をしきい値 θ として設定する．ここで，σ は異常度の標準偏差である．

4. 実験

本手法の有効性を確認するために大阪府立大学の職員が使用する端末と学生が使用する端末の Proxy ログを用いて不審な Web アクセスの検出実験を行った．

4.1 実験条件

大阪府立大学の職員は通常は端末を占有して使用しているため，個人ごとの固有の特徴が出現すると考え，職員の端末の Web アクセス履歴を正常な Web アクセスとして用いた．一方，学生が利用する端末は不特定多数の学生が利用しているため，様々な個人の特徴が出現すると考え，学生の Web アクセス履歴の一部を職員の端末の Web アクセス履歴に挿入し，不審な Web アクセスと見なして取り扱うこととした．2019 年 10 月 1 日 ~2019 年 11 月 30 日 (二ヶ月) の期間に収集した，大阪府立大学の職員の端末 30 台と学生用端末 50 台の Proxy ログを利用して実験を行った．端末ごとの Web アクセス履歴の 75%を学習用としきい値設定用に用い，25%をテスト用に用いた．75%の Web アクセス履歴中の 80%分を学習用に使用し，20%分をしきい値の設定用に用いた．職員の端末の Web アクセス履歴中には，不審な Web アクセスは存在しなかったため，テスト用 Web アクセス履歴に 50 台の学生の端末で収集した Web アクセス履歴を 1 台分ずつ等間隔に表れる様に挿入して，不審な Web アクセスとみなし，検出対象とすることとした．ここで，テスト用 Web アクセス履歴では，学習用 Web アクセス履歴には存在しないドメインにアクセスしていることがあり，このようなドメインは Word2vec で単語ベクトルに変換できないため，ドメインの遷移のユークリッド距離を算出できない．このような場合には，学習用 Web アクセス履歴中に現れたドメイン間の距離の最大値に距離の標準偏差の 3 倍を加えた値を存在しないドメインとの距離として用いることとした．使用する単語ベクトルの次元数は 100 次元とした．また，隠れ層の LSTM ユニットは 120 ユニットとした．評価方法は AUC 値（Area Under the Curve）と Precision，Recall,，F-measure，Accuracy を用いた．AUC 値は異常度のしきい値を変化させた際の False Positive Rate と True Positive Rate の割合に基づき描かれる ROC（Recieve Operating Charasteristic）曲線下の面積で定義され，値が 1 に近いほど完全に分類できていることを示す．Precision は，正と予測したデータのうち，実際に正であるものの割合である．Precision の算出式を次式に示す．

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

Recall は，実際に正であるデータのうち，正であると予測されたものの割合である．Recall の算出式を次式に示す．

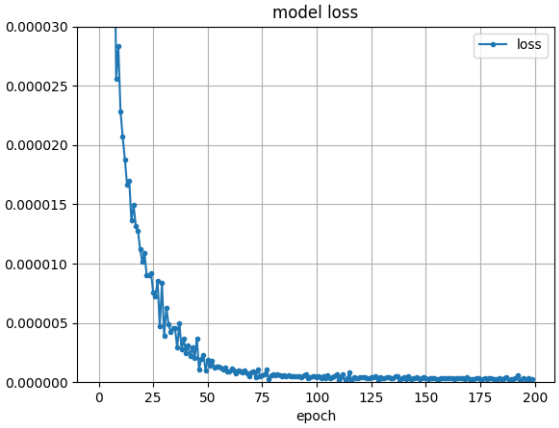

図 7　学習誤差の例
Fig. 7　Example of Training Error.

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

F-measure は Precision と Recall の調和平均である．F-measure の算出式を次式に示す．

$$F - \text{measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15}$$

Accuracy は，予測したデータのうち，実際に正しく判別できた割合である．Accuracy の算出式を次式に示す．

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

4.2 実験結果

ある職員の端末の Web アクセス履歴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 7 に示す．エポック数が 100 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．そこで，誤差の遷移の収束が確認できるエポック数 100 を以後の実験で使用するエポック数として設定した．端末ごとに Web アクセス履歴を学習し，異常を検出する実験を 30 端末分の Web アクセス履歴で行った．30 端末で調べた Precision，Recall，F-measure を表 2 に示す．表 2 では検出対象を正常とした時の値と異常とした時の値を示している．表 2 に示す結果より，AUC 値は 0.94 と高い値が得られ，本手法の有効性を確認できた．30 端末の AUC 値，Precision，Recall，Accuray の平均値と標準偏差を図 8 に示す．図 8 では，それぞれの値の標準偏差をエラーバーとして表示し，端末ごとに値の散らばりが存在しているかを確認した．異常な Web アクセスを適切に検知できた端末と検知できなかった端末との差を確認するために，AUC 値が高い 2 つの端末と低い 2 つの端末の AUC 値，Precision，Recall，Accuracy を図 8 に示している．

4.3 考察

表 2 に示す実験結果について考察する．検出対象を正常

表 2　実験の結果
Result of Experiment.

検出対象	AUC	Precision	Recall	F1-score
正常	0.94	0.99	0.98	0.99
異常		0.70	0.86	0.77

とした時と異常とした時のそれぞれで Recall は 0.98，0.86 と高い値を示している．このことから，異常度は，正常な Web アクセスのときに低く，異常な Web アクセスのときに高く算出され，しきい値 θ により正しく識別できていることがわかる．学生の端末の Web アクセス履歴は履歴ごとに特徴が異なっているが，それらを不審な Web アクセスとして正しく検出できたことから，本手法は攻撃手法に依存せず未知の不正通信を検知できると考えられる．

図 8 に示す端末ごとに算出した Accuracy，Precision，Recall の平均と標準偏差の結果について考察する．標準偏差が最も小さくなったのは Accuracy であることが確認できる．これは，職員の Web アクセス履歴が学生の Web アクセス履歴よりも多く，ほとんどの端末で職員の Web アクセス履歴を正常と識別できたためと考えられる．また，Recall，Precision の標準偏差が大きくなっている．

Precision と Recall の値が他の結果より低い device11 の Web アクセス履歴を確認した．device11 のテスト用 Web アクセス履歴に挿入した学生用 Web アクセス履歴中に，学習用 Web アクセス履歴に存在するドメインが含まれており，特徴ベクトルが類似したために，正常と誤識別していた．また，学習用 Web アクセス履歴に存在しないドメインに，職員がアクセスしていた場合，特徴ベクトルが類似しないために異常と誤識別している例もあった．それ以外にも受信データサイズや送信データサイズが学習用 Web アクセス履歴と大きく異なる場合の誤識別も含まれており，これらの要因により Precision，Recall の値が小さくなったと考えられる．

以上の問題に対しては，学習データを増加させる事により解決できると考えられる．今回の実験では 30 端末分，2 ヶ月間のデータを使用したが，端末数，期間ともにより大規模な実験を行って，適切な学習期間の推定などを行う予定である．また，今回の実験で利用したデータセットには異常が含まれていなかったため，学生の Web アクセス履歴を異常データとして挿入する事により実験を行ったが，本研究で検出を目指している標的型攻撃による Web アクセス履歴とはドメイン，送受信データサイズや UserAgent など記載されている内容が異なると考えられる．今後，標的型攻撃のデータセット等を利用して実験を行うことによって，本手法の有効性を確認したいと考えている．

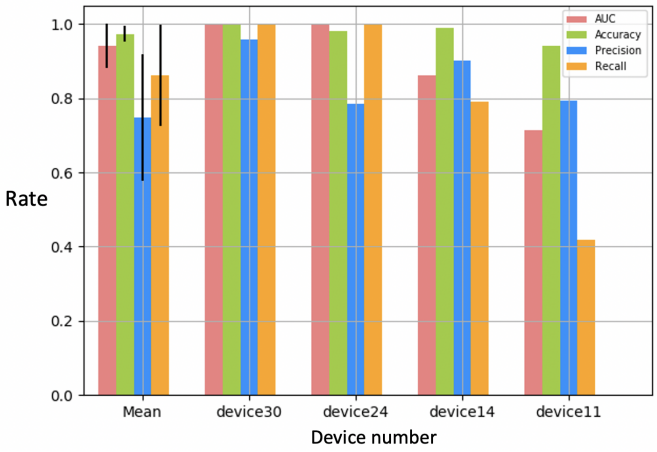

図 8　端末ごとの比較例（AUC and Accuracy and Precision and Recall）

Fig. 8　Examples of Comparison (AUC and Accuracy and Precision and Recall) of devices.

5. まとめ

本稿では，端末ごとの正常な Web アクセス履歴から抽出した特徴量を LSTM で学習する事で不審な Web アクセスを検出する手法を提案した．実験では，職員の Web アクセス履歴を学習し，不審な Web アクセスとみなして学生の Web アクセス履歴を検出する実験を行い有効性を確認した．検出対象のドメインが学習用データセットに含まれている場合や，送受信データサイズが類似している場合などに異常を正しく検出できない例があった．今後の課題としては，学習データ数の増加，送受信データサイズに頑健な特徴量の追加などが挙げられる．

参考文献

[1] 重田 真義，大谷 尚通：教師なし学習を活用したマルウェア感染検知システムの実装と評価，暗号と情報セキュリティシンポジウム (SCIS2018) 講演論文集，3F3-6（2018）．

[2] 三村 守，田中 秀摩：パラグラフベクトルへの Proxy サーバーログの丸投げ方式，コンピュータセキュリティシンポジウム (CSS2017) 講演論文集，Vol. 2017，No.2（2017）．

[3] Hongyu,L., Bo,L., Ming,L., et al. : CNN and RNN based payload classification methods for attack detection. Knowledge-Based Systems Vol163, pp.332-341（2019）．

[4] 神谷和憲，青木一史，中田健介，ほか：Firewall ログを用いたマルウェア感染端末の検知手法，情報処理学会第 77 回全国大会講演論文集，Vol. 2015，No.1，pp433-434（2015）．

[5] 江田 智尊，及川 孝徳，古川 和快，ほか：分散表現を用いたアラートログにおけるアノマリ検知，コンピュータセキュリティシンポジウム (CSS2019) 講演論文集，Vol.

2019，No.1，pp. 443-450（2019）.

[6] 三浦紘弥，三村守，田中秀磨：異常検知器を用いた未知の悪性マクロの検知手法，コンピュータセキュリティシンポジウム (CSS2018) 講演論文集，Vol. 2018，No.2，pp462-469（2018）.

[7] Congyuan,X., Jizhong,S., Xin, D., et al.: An intrusion detection system using a deep neural network with gated recurrent units, IEEE Access, Vol.6, pp.48697-48707（2018）.

[8] Hochreiter,S., Schmidhuber,J.: Long Short-Term Memory, Neural Computation, Vol. 9, No. 8, pp.1735-1780（1997）.