Computer Security Symposium 2017
23 - 25 October 2017

パラグラフベクトルへのプロキシサーバーログの丸投げ方式

三村 守[1]　田中 秀磨[1]

概要：サイバー攻撃の手段は刻々と進化しており，未知の不正通信の検知は大きな課題である．パターンマッチングや接続先のブラックリストは攻撃者に容易に回避され，未知のサイバー攻撃の検知には効果的ではない．そこで，ドライブバイダウンロード攻撃やC＆Cトラフィックの通信の特徴をとらえた多くの挙動ベースの検知手法が提案されている．しかしながら，多くの既存の手法はその攻撃手法に特化しており，その適応性は限られている．また，攻撃手法毎に特徴ベクトルを考える必要がある．本稿では，攻撃手法に依存せず、特徴ベクトルを考える必要がない汎用的な不正通信の検知手法を提案する．提案手法では，教師なし学習アルゴリズムで可変長のテキストから固定長の特徴ベクトルを学習するパラグラフベクトルを用い，通信ログの特徴を自動的に学習する．提案手法を MWS データセットの D3M および BOS に適用し，交差検証，時系列分析および交差データセット検証を実施した結果，プロキシサーバのログから高い精度で未知の不正通信を検知できることを確認した．

キーワード：ドライブバイダウンロード，C＆C，ニューラルネットワーク，Bag of Words，Word2vec，パラグラフベクトル，Doc2vec，サポートベクタマシン，ランダムフォレスト，多層パーセプトロン

Leaving All Proxy Server Logs to Pragraph Vector

Mamoru Mimura[1]　Hidema Tanaka[1]

Abstract: Cyber attack techniques are evolving every second, and detecting unknown malicious communication is a challenging task. Pattern-matching-based techniques and using malicious website blacklists are easily avoided, and not efficient to detect unknown malicious communication. Therefore, many behavior based detection methods are proposed, which use the characteristic of drive-by-download attacks or C&C traffic. However, many previous methods specialize the attack techniques and the adaptability is limited. Moreover, they have to decide the feature vectors every attack method. This paper proposes a generic detection method, which is independent of attack methods and does not need devising feature vectors. Our method uses Paragraph Vector an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts, such as sentences, paragraphs, and documents, and learns the context in proxy server logs. We conducted cross-validation, timeline analysis and cross-dataset validation with D3M and BOS in MWS datasets. The experimental results show our method can detect unknown malicious communication precisely in proxy server logs.

Keywords: Drive by Download, C&C, Neural Network, Bag of Words, Word2vec, Paragraph Vector, Doc2vec, Support Vector Machine, Random Forests, Multi-Layer Perceptron

1. はじめに

サイバー攻撃の脅威は年々深刻化しており，社会の関心は非常に高まっている．情報通信技術は急速に進化しており，それに伴ってサイバー攻撃の手法も刻々と進化してきた．2000 年代前半に流行したワームは影を潜め，近年では標的型メール攻撃やドライブバイダウンロード攻撃（ＤｂＤ攻撃）が主なサイバー攻撃の手法となっている．標的型メール攻撃やＤｂＤ攻撃により初期潜入が成功すると，Ｃ＆Ｃサーバを通じた遠隔操作により内部のネットワークが攻撃を受け，被害が深刻化することも珍しくな

[1] 防衛大学校
National Defense Academy

い．SOC (Security Operation Center) では IDS (Intrusion Detection System) のアラートやネットワーク機器のログを確認し，このようなサイバー攻撃の検知や対処を実施している．ネットワーク上でサイバー攻撃を検知する手法は，主にパターンマッチングとブラックリストに分類される．パターンマッチングは，攻撃の通信内容に固有のパターンを含んでいる場合に有効であり，IDS のシグネチャ等で活用されている．通信内容の固有のパターンは，固有の文字列や正規表現で定義される．しかしながら，近年では HTTP 等の標準のプロトコルを用いるだけでなく，正常通信を模倣し，固有の文字列や正規表現ではシグネチャの作成が困難な場合も増えてきている．このような場合には，ＤｂＤ攻撃で用いられるサーバやＣ＆Ｃサーバ等をブラックリストとして登録し，特定の接続先との通信を不正通信として検出する手法がとられることもある．しかしながらこの手法では，攻撃者はＤｂＤ攻撃に用いるサーバやＣ＆Ｃサーバ等を変更することで，容易に検知を回避することができる．また，あらかじめ接続先が判明していない場合には，不正通信を検知することができない．

このような課題に対応するため，ＤｂＤ攻撃やＣ＆Ｃトラフィックの通信の特徴をとらえ，通信挙動により未知の不正通信を検知する多くの手法が提案されてきた．しかしながら，多くの既存の手法はＤｂＤ攻撃やＣ＆Ｃトラフィックの通信固有の特徴をとらえているため，各々の攻撃にしか対応することができない．たとえば，一般にＤｂＤ攻撃の検知は入口対策，Ｃ＆Ｃトラフィックの検知は出口対策に分類されており，それぞれの攻撃の種類に応じた別々の対策技術として検討されることが多い．このような手法では，攻撃の種類が異なる場合や，攻撃の手法が変化した場合には対応することができない．細部についても同様に，ＤｂＤ攻撃の状態遷移や Exploit Kit の特徴，Ｃ＆Ｃトラフィックの挙動等，様々な特徴をうまくとらえた手法が提案されている．このような手法では，攻撃の特徴が変化した場合には対応することができない．また，検知に利用する攻撃の特徴は，人間が検討する必要がある．

本稿では，攻撃の手法や特徴の変化に対応するために，ニューラルネットワークの自動的に特徴を学習する点に着目する．ニューラルネットワークの技術は，画像認識や音声認識をはじめ，自然言語処理においてもすぐれた分類性能を発揮している．特に近年では，単語の意味を考慮して文章をベクトルに変換するモデルが提案されたため，単語の出現頻度だけでなく，その意味や前後関係を考慮した分類が可能となってきた．本稿では，プロキシサーバに記録されたログを自然言語として解釈し，ニューラルネットワークを用いて不正通信と正常通信の特徴を自動的に学習させることを試みる．そして，自動的に学習した特徴を教師あり学習モデルに入力し，不正通信と正常通信を分類する手法を提案する．提案手法は攻撃の手法に依存せず、特徴を抽出する必要がない汎用的な不正通信の検知手法である．さらに，提案手法を MWS データセットの D3M および BOS に適用し，交差検証，時系列分析および他のデータセットを用いた交差データセット検証を実施する．

以下，第２節では関連研究について示し，提案手法と比較してその違いを明確にする．第３節では，文章をベクトルに変換するための自然言語処理技術について説明する．第４節では，自然言語処理技術を用いて自動的に作成した特徴ベクトルを教師あり学習モデルに入力し，不正通信と正常通信を分類する提案手法について説明する．第５節では提案手法を MWS データセットの D3M および BOS に適用し，交差検証，時系列分析および他のデータセットを用いた交差データセット検証を実施する．第６節では実験結果を考察し，最後にまとめと今後の課題を示す．

2. 関連研究

本稿では，プロキシサーバのログからパターンマッチングやブラックリストを用いずに，不正通信を検知することを目的としている．これまでに提案された通信挙動の特徴を用いる手法の多くは，パケット単位の情報へのアクセスを前提としている．しかしながら，現実的にはパケット単位の情報をすべて保存している組織は少ない．プロキシサーバのログから不正通信を検出する手法であれば，多くの組織に適用することが可能である．

文献 [1] では，URI を共通のリソースを持つ PATH 毎に分類し，クエリ文字列に含まれるパラメータから得られる特徴を学習する統計的モデルを構築し，異常を検知する手法を提案している．この手法は，Web サーバに対するバッファオーバフロー，ディレクトリトラバーサル，クロスサイトスクリプティング等の直接的な攻撃を検知することを想定している．われわれの提案手法では，機械学習を不正通信と正常通信の２値分類のために用いる．また，ＤｂＤ攻撃やＣ＆Ｃトラフィックの検知も想定している．

文献 [2] では，URL の文字列に含まれるドメイン，PATH 等から得られる情報を用いて特徴ベクトルを作成し，機械学習により不正 URL と正常 URL を分類する手法を提案している．この手法では，プロキシサーバのログから得られる情報の他にも，リンクの人気度，Web ページのコンテンツ，DNS およびトラフィック量から得られる特徴ベクトルも利用している．われわれの提案手法では，プロキシサーバのログから得られる情報のみを利用しており，外部の情報や通信内容から得られる情報を必要としない．また，特徴ベクトルを考える必要はない．

文献 [3] では，URL の文字列に含まれるホスト名，トップレベルドメイン，PATH 等から特徴ベクトルを作成し，オンライン学習のアルゴリズムを用いて不正 URL と正常 URL を分類する手法を提案している．この手法では，URL をドット（.），スラッシュ（／），クエスチョンマーク（？），

イコール（=），アンド（&）等の区切り文字でトークンに分離している．われわれの提案手法では，自然言語処理技術を適用するために類似の手法を用いる．またこの手法では，プロキシサーバのログから得られる情報の他にも，IP アドレスの登録情報，Whois の登録情報，ドメイン名の登録情報，ブラックリスト，地理的特性，接続速度等のホストから得られる情報も用いている．われわれの提案手法では，プロキシサーバのログから得られる情報のみを利用する．また，特徴ベクトルを考える必要もない．

文献 [4] では，URL の構造，特徴的あるいはブランド名等の文字列から特徴ベクトルを作成し，機械学習を用いてフィッシングに用いられる URL を検知する手法を提案している．この手法では，フィッシングに用いられる URL のみを対象としている．われわれの提案手法が対象とする不正通信には，ＤｂＤ攻撃やＣ＆Ｃサーバに用いられる URL も含まれており，特に制約はない．また，特徴ベクトルを考える必要もない．

文献 [5] では，人間が正誤を判断してラベルを付与するコストを問題視し，不正 URL を検知するための分類器を，自動で更新するオンラインアクティブ学習のフレームワークを提案している．この手法では，分類器の更新のために Whois の登録情報，ドメイン名の登録情報，ブラックリスト等を用いている．われわれの提案手法では，プロキシサーバのログから得られる情報のみを利用し，外部のサイトから得られる情報を必要としない．

文献 [6] では，IP アドレス，ドメイン名，FQDN、URL，PATH，ファイル等からネットワークグラフを作成し，特定の相関関係を持つ主体に着目することで，大規模ネットワークからマルウェアのダウンロードを検知する手法を提案している．この手法では，プロキシサーバのログから得られる情報を用いているが，多くのホストとインターネット間の通信を傍受する必要があり，ISP 等の大規模なネットワークで運用を想定している．また，ダウンロードされたファイルの種類も必要である．われわれの提案手法は，ネットワークの規模に依存せずに運用することが可能であり，ダウンロードされたファイルの種類も必要としない．

文献 [7] では，HTTP の Request および Response から Location，Referrer 等を抽出して閲覧経路のグラフを構築し，ＤｂＤ攻撃の経路をさかのぼることができるシステムを提案している．このシステムでは，不正サイトの識別のために，経路のホップ数，ドメインの継続期間，ドメイン名の共通性等を利用している．われわれの提案手法では，プロキシサーバのログから得られる情報のみを利用し，外部のサイトから得られる情報を必要としない．また，ＤｂＤ攻撃以外の不正通信も検知することが可能である．

文献 [8] では，プロキシサーバのログをフローに分類し，URL，PATH，クエリ，ファイル名等から様々な特徴を抽出し，不正通信と正常通信を分類するための最終的な特徴ベクトルを自動的に作成する手法を提案している．この手法では，特徴ベクトルを自動的に作成することが可能であるが，そのベースとなる特徴を考える必要がある．われわれの提案手法では，ベースとなる特徴すら考える必要はない．

文献 [9] では，プロキシサーバのログを FQDN 毎に分類し，標的型攻撃に用いられる RAT (Remote Access Trojan or Remote Administration Tool) のＣ＆Ｃトラフィックの特徴を抽出し，機械学習により不正通信と正常通信を分類する手法を提案している．この手法では，RAT が同一の PATH に規則的に継続してアクセスする特徴を利用している．われわれの提案手法では，Ｃ＆Ｃトラフィック以外の不正通信も検知することが可能である．また，特徴ベクトルを考える必要もない．

3. 自然言語処理技術

3.1 Bag-of-Words (BoW)

自然言語をコンピュータで処理するためには，文章や単語をベクトルに変換する必要がある．文章をベクトルに変換する最も基本的なモデルとしては，Bag-of-Words (BoW) が挙げられる．BoW は，文章中の単語の出現頻度を数え，その出現回数をその単語に相当するベクトルの要素の値に変換するモデルである．BoW においては，文章のベクトルの値は単語の出現頻度で表現されるため，同一の文章である場合を除外すると，ベクトル同士の演算結果は意味をなさない．つまり，BoW では単語の出現回数は表現されるが，その順序や意味は表現されない．

3.2 Word2vec

2013 年になると，単語の意味を表現する Word2vec[10] というニューラルネットワークのモデルが提案された．Word2vec では，文章の中で単語同士が交換可能であるかどうかに着目し，単語の意味を考慮した分散表現を実現している．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であり，単語の語学上のコンテキストを構成するように訓練される．Word2vec は大量のコーパスを入力とし，数 100 次元のベクトル空間を構築する．そのベクトル空間においては，共通のコンテキストを持つ単語は，おおよそ近い位置に配置される．これは，コーパス内である単語の周辺に共起する単語の確率に基づいている．Word2vec のアルゴリズムには，周辺の単語から中心の単語を予測する Continuous Bag-of-Words (CBoW) と，中心の単語から周囲の単語を予測する Skip-Gram がある．Word2vec のモデルを用いることで，単語のベクトル演算により単語の類似度を求めたり，類似する単語を推定することが可能となる．

3.3 Doc2vec

Word2vec は，単語の意味を考慮した分散表現を実現したモデルであった．この考え方を文章全体に拡張し，文章を単語の集合としてベクトルを割り当てる手法が提案された [11]．この拡張は，Doc2vec あるいは Paragraph2vec とも呼ばれる．Doc2vec のアルゴリズムには，Word2vec の CBoW の応用である Distributed Memory (DM) と，Skip-gram の応用である Distributed Bag-of-words (DBoW) がある．DM は CBoW の考え方を応用したものであり，入力ベクトルに単語列だけでなく，文章の ID を追加したものである．DBoW は Skip-gram の考え方を応用したものであり，入力を単語から文章の ID に変更したものである．Doc2vec のモデルを用いることで，文章のベクトル演算により文章の類似度を求めたり，文章のベクトル表現を推定することが可能となる．

4. 提案手法

4.1 プロキシサーバのログの分かち書き

提案手法では，プロキシサーバのログを自然言語と解釈してコーパスを作成する．ログからコーパスを作成するためには，意味のある単語に分かち書きを実施する必要がある．多くのプロキシサーバのログには，アクセスが発生した時刻，クライアントの IP アドレス，HTTP ステータスコード，リクエストの内容（メソッドおよび URL を含む.），クライアントが受信したデータのサイズ，クライアントが使用する User Agent が記録されている．ここでいうクライアントとは，外部のサイトを閲覧する利用者の端末のことである．プロキシサーバのログは，内部の端末から外部のサイトへの要求を起点とし，外部のサイトからの応答を併せて 1 件として時系列で記録される．

提案手法ではプロキシサーバのログを，HTTP ステータスコード，リクエストの内容，クライアントが受信したデータのサイズ，クライアントが使用する User Agent に分離する．さらに，リクエストの内容はメソッド，URL およびプロトコルのバージョンに分離する．

4.2 URL の分かち書き

URL は不正通信を識別するための最も重要な項目であると考えられるため，さらに詳細に分かち書きを実施する．URL の分かち書きの例を**図 1** に示す．

まず，スキーム，FQDN (Fully Qualified Domain Name) および PATH 以降に分離する．次に，FQDN をドット（.）で分離する．これにより，トップレベルドメインの国や組織，サーバの用途に応じたサブドメイン名等が単語として区別できるようになる．PATH はスラッシュ（／）およびドット（.）で分離し，クエスチョンマーク（?）以降のパラメータの部分はイコール（=）およびアンド（&）で分離する．これにより，PATH からはディレクトリ名，ファイル名および拡張子が単語として得られる．パラメータの部分からは，サーバで動作するプログラムで使用される変数名やその値が得られる．ここで，一度しか使用されない変数の値はコーパスとして使用しない．なお，分離に使用したドット（.），スラッシュ（／），クエスチョンマーク（?），イコール（=）およびアンド（&）の記号の意味も考慮するため，単語として残すこととした．特に，スラッシュ（／）は URL の構造に密接に関係しているため，その構造を特徴として自動的とらえることが期待できる．

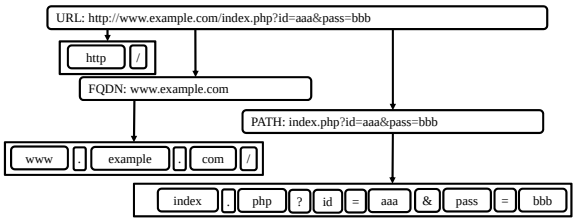

図 1 URL の分かち書きの例
Fig. 1 An example of a URL leaving a space between words.

以上のような分かち書きにより，プロキシサーバのログを単語に分離してコーパスを作成する．

4.3 提案手法

提案手法の概要を**図 2** に示す．まず，前述の手法で分かち書きした正常通信および不正通信の既知のログからコーパスを作成し，文章のベクトル表現のモデルを構築する．文章のベクトル表現を実現するモデルは，BoW や Doc2vec を想定している．次に，モデルの構築に用いた正常通信および不正通信のログをそのモデルに入力し，訓練ベクトルを作成する．さらに，その訓練ベクトルにラベルを付与して別のモデル（分類器）に入力して訓練を実施する．この別のモデルは，サポートベクタマシン (SVM)，ランダムフォレスト (RF) 等の分類に用いることができる教師あり学習モデルを想定している．最後に，検査対象とする未知のログからテストベクトルを作成し，モデル（分類器）に入力してラベルを得る．出力されるラベルは，正常通信あるいは不正通信のどちらかとなる．以上が提案手法の概要である．

4.4 実装

提案手法を Python-2.7 を用いて試験プログラムとして実装した．文章のベクトル表現は，BoW および Doc2vec を gensim-1.01[12] を用いて実装した．Doc2vec の次元数は 100，学習回数は 30 回とし，アルゴリズムは DBoW を選択した．分類器は SVM，RF に加え，多層パーセプトロン (MLP) を実装した．SVM および RF の実装には Scikit Learn-0.18.1[13] を用いた．MLP の実装には Chainer-1.23[14]，CUDA 8.0 および cuDNN-8.0 を用いた．

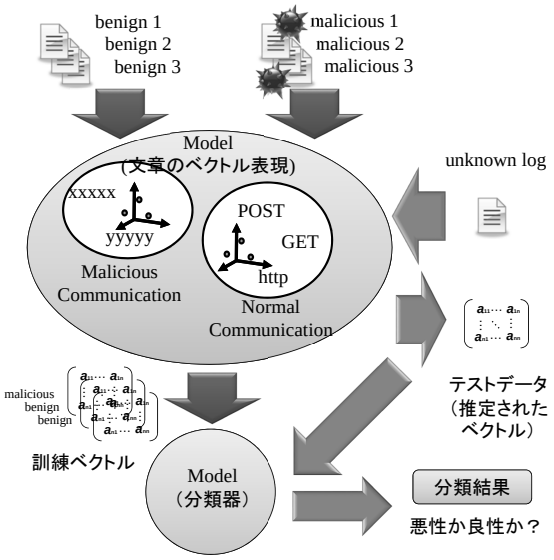

図 2 提案手法の概要
Fig. 2 An outline of the proposed method.

MLP は 3 層構成とし，入力ユニット数をテストデータの次元数，中間ユニット数を 500，出力ラベル数を 2 とした．1 〜 2 層の活性化関数は ReLU (Rectified Linear Unit) を用い，dropout は使用しない．出力層にはソフトマックス関数を用い，損失関数として以下の交差エントロピー誤差を用いた．

$$E = -\sum_{n}^{N}\sum_{i}^{D} t_{ni} \log y_i$$

式中の N はデータの数，D は出力数のユニット数，y は出力された値，t は真のラベルの値を示す．ミニバッチ勾配降下法のバッチサイズは 100 とし，最適化アルゴリズムは Adam (Adaptive Moment Estimation) を用いた．

5. 検証実験

5.1 実験環境

提案手法の検知精度を評価するため，検証実験を実施する．試験プログラムが動作する実験環境を**表 1** に示す．

表 1 実験環境

CPU	Core i7-5820K 6core 3.3GHz
Memory	DDR4 SDRAM 24GB
GPU	GeForce GTX980/4G
OS	Windows-8.1

5.2 実験内容

検知精度の評価には，MWS datasets[15] に含まれている D3M 2010〜2015 (D3M)，BOS 2014〜2016 (BOS) および NCD in MWSCup 2014 (NCD) を用いる．D3M にはＤｂＤ攻撃，BOS にはＣ＆Ｃトラフィックの不正通信の pcap ファイルが含まれている．NCD には正常通信の pcap ファイルが含まれている．提案手法ではプロキシサーバのログからモデルを構築することを想定している．そのため，これらの pcap ファイルから HTTP の通信内容を抽出し，Request とそれに対応する Response から擬似ログを作成した．D3M，BOS および NCD から作成した擬似ログを正例と負例が均等となるように合成し，10 分割交差検証および時系列分析を実施する．コーパスは訓練データのみから作成し，テストデータは使用しない．識別器に入力するためのベクトルは，1 件のログおよび複数件のログから作成して精度を比較する．1 件のログから 1 つのベクトルを作成する場合には，1 件のログを 1 つの文章として取り扱う．複数件のログから 1 つのベクトルを作成する場合には，単純に連続する 10 件のログを 1 つの文章として取り扱う．時系列分析においては，前半の部分を訓練データとし，以後の残りの部分をテストデータとする．

さらに，交差データセット検証のため，MALWARE-TRAFFIC-ANALYSIS.NET[16] からダウンロードした Exploit Kit の通信データを追加で用いる．この通信データには，2014 年から 2017 年の間に収集した約 1 G の pcap ファイル (MTA) が含まれている．交差データセット検証では，D3M および MTA を相互に訓練データおよびテストデータとし，連続する 10 件のログを 1 つの文章としてその精度を確認する．

5.3 評価指標

本実験では，評価指標として Precision (P)，Recall (R) および F 値 (F) を用いる．各評価指標の定義は以下の数式のとおりである．

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

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

$$F - measure = \frac{2Recall \times Precision}{Recall + Precision}$$

式中の TP (True Positive)，FP (False Positive)，TN (True Negative) および FN (False Negative) の関係は**表 2** に示すとおりである．

表 2 クラス分類の結果
Table 2 Confusion matrix for two possible outcomes.

		真の値	
		正	負
予測結果	正	TP	FP
	負	FN	TN

表 3　10 分割交差検証の結果 (D3M)
Table 3　The result of the 10-fold cross-validation (D3M).

log	モデル	分類器	NCD			D3M		
			P	R	F	P	R	F
1	BoW	SVM	0.57	0.91	0.70	0.75	0.30	0.41
		RF	0.57	0.91	0.70	0.74	0.30	0.41
		MLP	0.59	0.91	0.71	0.77	0.34	0.44
	Doc2vec	SVM	0.92	0.89	0.90	0.90	0.92	0.91
		RF	0.90	0.93	0.91	0.92	0.89	0.90
		MLP	0.94	0.92	0.93	0.92	0.94	0.93
10		SVM	1.00	0.96	0.98	0.96	1.00	0.98
		RF	0.98	0.97	0.97	0.97	0.98	0.98
		MLP	0.99	0.98	0.98	0.98	0.99	0.98

表 4　10 分割交差検証の結果 (BOS)
Table 4　The result of the 10-fold cross-validation (BOS).

log	モデル	分類器	NCD			BOS		
			P	R	F	P	R	F
1	BoW	SVM	0.94	0.98	0.96	0.98	0.93	0.96
		RF	0.94	0.98	0.96	0.98	0.93	0.96
		MLP	0.94	0.98	0.96	0.98	0.94	0.96
	Doc2vec	SVM	0.94	0.97	0.95	0.97	0.94	0.95
		RF	0.97	1.00	0.98	1.00	0.97	0.98
		MLP	0.98	0.99	0.98	0.99	0.97	0.98
10		SVM	1.00	1.00	1.00	1.00	1.00	1.00
		RF	0.98	1.00	0.99	1.00	0.98	0.99
		MLP	1.00	1.00	1.00	1.00	1.00	1.00

5.4　実験結果

まず，10 分割交差検証の結果を**表 3** および**表 4** に示す．D3M においては，BoW を用いた場合の精度は低く，Doc2vec を用いた場合の精度は良好であった．BOS においては，BoW を用いた場合にも Doc2vec を用いた場合にも精度は良好であった．これは，BOS のC＆C トラフィックには同じ URL に継続してアクセスするという特徴があり，BoW がその出現回数を特徴としてとらえたためであると考えられる．10 分割交差検証の結果，Doc2vec を用いた提案手法では，D3M および BOS の不正通信を高精度で検知できることが確認できた．

次に，時系列分析の結果を**表 5** および**表 6** に示す．D3M においては，BoW を用いた場合の精度はさらに低下し，Doc2vec を用いた場合の精度はやや低下したものの，比較的に良好であった．BOS においては，BoW を用いた場合の精度は良好であった．これは，BoW が同じ URL の出現回数を特徴としてとらえたためであると考えられる．Doc2vec を用いた場合の精度は，1 件のログにおいては低下がやや顕著であったが，複数件のログの場合には高い精度が維持された．特に，複数件で MLP の場合の精度は良好であった．これは，ログのコンテキストが考慮されたためであると考えられる．

表 5　時系列分析の結果 (D3M)
Table 5　The result of the timeline analysis (D3M).

log	モデル	分類器	NCD			D3M		
			P	R	F	P	R	F
1	BoW	SVM	0.54	0.91	0.67	0.71	0.22	0.34
		RF	0.53	0.90	0.67	0.70	0.22	0.34
		MLP	0.54	0.90	0.67	0.68	0.22	0.33
	Doc2vec	SVM	0.92	0.84	0.88	0.85	0.92	0.89
		RF	0.92	0.87	0.90	0.89	0.93	0.91
		MLP	0.92	0.87	0.92	0.89	0.97	0.93
10		SVM	1.00	0.96	0.98	0.96	1.00	0.98
		RF	0.97	0.96	0.96	0.96	0.97	0.96
		MLP	0.99	0.99	0.99	0.99	0.99	0.99

表 6　時系列分析の結果 (BOS)
Table 6　The result of the timeline analysis (BOS).

log	モデル	分類器	NCD			BOS		
			P	R	F	P	R	F
1	BoW	SVM	0.93	0.99	0.96	0.99	0.92	0.95
		RF	0.93	0.99	0.96	0.99	0.92	0.95
		MLP	0.93	0.99	0.96	0.99	0.92	0.95
	Doc2vec	SVM	0.77	0.96	0.86	0.95	0.71	0.81
		RF	0.74	1.00	0.85	1.00	0.64	0.78
		MLP	0.76	0.97	0.85	0.95	0.69	0.80
10		SVM	0.96	1.00	0.98	0.99	0.96	0.98
		RF	0.79	1.00	0.88	1.00	0.69	0.82
		MLP	0.95	1.00	0.98	1.00	0.95	0.98

表 7　交差データセット検証の結果
Table 7　The result of the cross-dataset-validation.

Train	Test	分類器	NCD			D3M or MTA		
			P	R	F	P	R	F
MTA	D3M	SVM	0.97	0.91	0.94	0.92	0.97	0.94
		RF	0.80	0.95	0.87	0.94	0.76	0.84
		MLP	0.94	0.96	0.95	0.96	0.94	0.95
D3M	MTA	SVM	0.96	0.96	0.96	0.96	0.96	0.96
		RF	0.86	0.96	0.91	0.96	0.85	0.90
		MLP	0.92	0.97	0.94	0.97	0.92	0.94

最後に，交差データセット検証の結果を**表 7** に示す．複数件のログの場合には，交差データセット検証においても高い精度が維持された．これは，D3M および MTA がともにＤｂＤ攻撃の通信内容を含み，ログのコンテキストが類似していたためであると考えられる．

6.　考察

6.1　正確性

検証実験の結果，提案手法ではプロキシサーバのログのみを用いているにもかかわらず，高い精度で不正通信

と正常通信の区別が可能であることが確認できた．特に，Doc2vec を用いた場合の精度は良好であり，交差検証においては 1 件のログから区別することさえ可能であった．これは，Doc2vec が 1 件のログの不正通信と正常通信の特徴を自動的にとらえ，適切なログの分散表現を構築したためであると考えられる．この結果は，ＤｂＤ攻撃に用いられる URL と通常の URL には違いがあることを意味している．提案手法では URL を単語に分離し，Doc2vec は一定の範囲内のそれらの単語間の共起確率の割合を表現する．しかしながら，単にその単語の出現回数のみを表現する BoW では，1 件のログから区別することはできなかった．したがって，区別に用いられた違いは，URL を構成する単語自身と URL の構造であると考えられる．いくつかの従来の手法は，この URL の構造を不正通信を検知するために利用する．しかしながら，その URL の構造は人間がどのように抽出するかを明示する必要があった．提案手法では，明確な人間の指示がなくても URL の構造を自動的に学習する．時系列分析においては，1 件のログにおいては精度の低下はやや顕著であった．特に，その低下は BOS において最も顕著であった．これは，BOS のＣ＆Ｃトラフィックの URL は正常通信の URL と似ており，やや区別が難しいことを示している．たとえば，Emdivi は正常通信を模倣するため，標準の HTTP や改ざんした Web サイトをＣ＆Ｃサーバとして悪用する．しかしながら，複数件のログにおいては高い精度を維持することが可能であった．これは，Doc2vec が 1 件のログの特徴だけでなく，ログのコンテキストを加味した適切なログの分散表現を構築したためであると考えられる．また，その精度は交差データセット検証においても維持されることを確認することができた．このように，提案手法では 1 件のログやログのコンテキストの特徴を自動的にとらえ，不正通信と正常通信を高精度で区別することが可能であると考えられる．

誤検知の主な原因は，認証やストリーミング等の Web API を提供するサイトであった．これらのサイトでは多くのパラメータを使用するため，Exploit Kit やＣ＆Ｃトラフィックと挙動が類似しているものがある．更新プログラムのダウンロードやウイルス対策ソフトのアップデートについても，やや誤検知が発生した．これらは，マルウェアのダウンロードや感染の挙動に類似している．また，不正通信のログに含まれているわずかな正常通信も誤検知の原因となった．これは，ある意味では誤検知とは言えないが，不正通信のログの純度を高めることにより，精度を改善することが可能である．

6.2 適応性

提案手法では，共通の手法を用いてＤｂＤ攻撃とＣ＆Ｃトラフィックを不正通信として区別することが可能である．ＤｂＤ攻撃であれＣ＆Ｃトラフィックであれ，分かち書きしたログさえ入力すれば，攻撃の種類を考慮することなく提案手法を適用することが可能である．提案手法では，ＤｂＤ攻撃の状態遷移や Exploit Kit の特徴，Ｃ＆Ｃトラフィックの挙動のような，攻撃の特徴を考慮する必要もない．単に不正通信のログにラベルを付与して入力するだけで，その不正通信の特徴は自動的に学習される．従来の機械学習を用いた手法のように，特徴ベクトルを考える必要もない．このように提案手法は，攻撃の種類が異なる場合や変化した場合にも柔軟に対応することが可能であり，高い適応性をもっていると言える．

6.3 持続性

提案手法は，ニューラルネットワークを用いて自動的に不正通信と悪性通信の違いを学習する．一般にニューラルネットワークでは，どの特徴が最も違いを表現したかを明確にすることが難しい傾向がある．これを逆に考えると，攻撃者にとってもその特徴を知ることは困難であると言える．われわれはその特徴を特定するために，FQDN，User Agent 等のいくつかの要素を除外して同様の実験を実施したが，その結果に顕著な違いは認められなかった．この結果は，提案手法は FQDN 等の特定の要素に依存していないことを意味している．したがって，攻撃者が提案手法を回避する有効な選択肢はないものと考えられる．攻撃者が提案手法を回避するためにとり得る唯一の選択肢は，正常通信を限りなく模倣することである．ゆえに，提案手法は効果的かつ持続性があるものと考える．

6.4 実用性

今回の実験では，Doc2vec と分類器の学習には数分程度を要し，予測には数秒程度を要した．学習に要する時間は，ログの量が増加するほど長くなる．しかしながら，検知システムを構築する場合には，学習を事前に済ますことが可能である．提案手法では，1 件あるいは 10 件のログのみから不正通信を検知することが可能である．したがって，提案手法はリアルタイムでネットワークトラフィックやプロキシサーバのログを分析する用途に適用可能である．交差データセット検証においては，他のデータセットに対しても相互に高い精度が維持されることを確認することができた．この結果から，提案手法は実際の未知のデータを分析する用途においても有効であると考えられる．提案手法では，入力として不正通信と正常通信の通信データを必要とする．これらのデータは，不正通信のデータを公開しているサイトや自組織のプロキシサーバのログ等から容易に入手することが可能である．また，標準ログフォーマットに含まれる項目のみを使用し，クライアントの IP アドレスも必要としない．したがって，提案手法を適用するための要件は少ないため，その実用性は非常に高いと考えられる．

6.5 倫理

提案手法が入力として必要とする通信データには，プライバシーや通信の秘密に関係するデータが含まれる可能性がある．特に、ペイロードや，通信ログに含まれるクライアントの IP アドレスは，個人の識別に結びつく可能性があるため注意が必要である．多くの既存の手法は，パケット単位の情報へのアクセスを前提としている．これらの手法では，ペイロードにアクセスする可能性を排除することが難しい．これに対し，提案手法ではパケット単位の情報へのアクセスを必要としない．さらに，クライアントの IP アドレスも使用せず，その区別すらも必要としない．また，提案手法を未知のデータを分析する用途に用いる場合には，ペイロードや通信ログそのものを必要としない．未知のデータを分析する用途においては，あらかじめ構築した学習済みのモデルのみを必要とする．この学習済みのモデルには，ペイロードや通信ログそのものは含まれないため，組織間での共有や公開も現実的に可能であると考えられる．したがって，提案手法は倫理的側面においても相対的に優れていると言える．

7. おわりに

本稿では，プロキシサーバのログからコーパスを作成して文章をベクトルに変換するモデルを構築し，得られたベクトルを教師あり学習モデルに入力して不正通信と正常通信を分類する手法を提案した．さらに，提案手法を実装し，MWS データセットの D3M および BOS，MALWARE-TRAFFIC-ANALYSIS.NET から入手した通信データ (MTA) を用いて交差検証，時系列分析および交差データセット検証を実施した．その結果，提案手法は高精度で未知の不正通信を検知できることを示した．また，１件のログだけでなく，ログのコンテキストの特徴を自動的にとらえる用途において，Doc2vec は極めて有効であることを示した．提案手法では，攻撃の種類に依存せずにあらゆる不正通信に対応することが可能であると考えられる．また，攻撃や特徴や特徴ベクトルを考慮する必要もない．適用するための要件も少なく，高い実用性をもっている．

今後の課題としては，提案手法の実運用のネットワークへの適用が挙げられる．今回の検証実験では，MWS datasets に含まれている D3M 2010〜2015 (D3M)，BOS 2014〜2016 (BOS)，NCD in MWSCup 2014(NCD) および MALWARE-TRAFFIC-ANALYSIS.NET から入手した通信データ (MTA) を用いた．提案手法に実運用のネットワークの通信を用いた場合の精度については，若干の検証の余地がある．提案手法を用いたリアルタイムの不正通信の検知システムの構築についても今後の課題である．今回はプロキシサーバのログを自然言語として解釈し，Doc2vec に入力してその特徴を自動的に表現することに成功した．同様に IDS のアラート，セキュリティ器材，あるいは SIEM (Security Information and Event Management) のログを自然言語として解釈することができれば，さらに詳細に攻撃の内容を表現することができる可能性がある．他のログやトラフィックへの応用についても今後の課題である．

参考文献

[1] Kruegel, C., and Vigna, G.: *Anomaly Detection of Web-based Attacks* Proc. 10th ACM Conference on Computer and Communications Security, pp.251-261 (2003).

[2] Choi, H., Zhu, B.B., and Lee, H.: *Detecting Malicious Web Links and Identifying Their Attack Types* Proc. 2nd USENIX Conference on Web Application Development, pp.1-11 (2011).

[3] Ma, J., Saul, L.K., Savage, S., and Voelker, G.M.: *Learning to Detect Malicious URLs* ACM Trans. on Intelligent Systems and Technology, Vol.2, No.3, Article 30 (2011).

[4] Huang, H., Qian, L., and Wang, Y.: *A SVM-based Technique to Detect Phishing URLs* Information Technology Journal, Vol.11, No.7, pp.921-925 (2012).

[5] Zhao, P., and Hoi, S.C.: *Cost-sensitive Online Active Learning with Application to Malicious URL Detection* Proc. 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.919-927 (2013).

[6] Invernizzi, L., Miskovic, S., Torres, R., Saha, S., Lee, S., Mellia, M., Kruegel, C., and Vigna, G.: *Nazca: Detecting Malware Distribution in Large-scale Networks* Proc. Network and Distributed System Security Symposium, (2014).

[7] Nelms, T., Perdisci, R., Antonakakis, M., and Ahamad, M.: *Webwitness: Investigating, Categorizing, and Mitigating Malware Download Paths* Proc. 24th USENIX Security Symposium, pp.1025-1040 (2015).

[8] Bartos, K., and Sofka. M.: *Optimized Invariant Representation of Network Traffic for Detecting Unseen Malware Variants* Proc. 25th USENIX Security Symposium, pp.806-822 (2016).

[9] Mimura, M., Otsubo, Y., Tanaka, H., and Tanaka, H.: *A Practical Experiment of the HTTP-Based RAT Detection Method in Proxy Server Logs* Proc. 12th Asia Joint Conference on Information Security, (2017).

[10] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., and Dean, J.: *Distributed Representations of Words and Phrases and Their Compositionality*, In Advances in Neural Information Processing Systems, pp.3111-3119, (2013).

[11] Le, Q., and Mikolov, T.: *Distributed Representations of Sentences and Documents* Proc. 31st International Conference on Machine Learning, pp.1188-1196, (2014).

[12] gensim (online)，入手先 ⟨https://radimrehurek.com/gensim/⟩ (2017.06.26).

[13] scikit-learn (online)，入手先 ⟨http://scikit-learn.org/⟩ (2017.06.26).

[14] Chainer (online)，入手先 ⟨https://chainer.org/⟩ (2017.06.26).

[15] 神薗 雅紀，秋山 満昭，笠間 貴弘，村上 純一，畑田 充弘，寺田 真敏：マルウェア対策のための研究用データセット 〜MWS Datasets 2015〜，情報処理学会研究報告，Vol.2015-CSEC-70, No.6 (2015).

[16] MALWARE-TRAFFIC-ANALYSIS.NET (online)，入手先 ⟨http://www.malware-traffic-analysis.net/⟩ (2017.06.26).