Vol.2019-IOT-44 No.4
2019/3/7

Doc2vecによるパラグラフベクトルを用いたマルウェア感染端末の検出

南 辰典[1,a)]　青木 茂樹[1,b)]　宮本 貴朗[1,c)]

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困難なため，マルウェアに感染した後の感染端末の早期発見が重要視されている．感染端末は感染前と感染後では異なる通信をするため，感染前の端末の通信ログを学習することで，動きの違いを検知できると考えられる．そこで本稿では，感染前の端末の通信ログを学習し，不審な動きを検出する手法を提案する．本手法では，端末毎のFirewallログとProxyログにDoc2vecを適用してパラグラフベクトルを抽出し，そのベクトルをOne-Class SVMで学習して外れ値を検出することで，不審な動きを検出する．

キーワード：標的型攻撃，パラグラフベクトル，Doc2vec，One-Class SVM，自然言語処理技術

Detection of malware infected host using paragraph vector by Doc2vec

Tatsunori Minami[1,a)]　Shigeki Aoki[1,b)]　Takao Miyamoto[1,c)]

Abstract: APT(Advanced Persistent Threat) attacks has become serious problem in the world. Attackers infect targeted hosts with malware and remotely control them. The Attackers investigate network to which the infected hosts belong using their hosts before stealing confidential information. The infected hosts that operated by the attackers behave differently than before malware infection. Such behaviors are recorded to network log. Thus, we thought that we can detect different behaviors by learning the network log of before infection. In this paper, we propose a method to detect malware infected hosts by learning the behaviors from the network log of the hosts before infection. In this method, we learn behaviors before infection from firewall log and proxy server log using Doc2vec. Further, we detect malware infected hosts by learning the vectors using One-Class SVM and detecting outliers.

Keywords: APT attack, Paragraph Vector, Doc2vec, One-class SVM, Natural Language Processing Technique

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，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）などの侵入防御装置を設置し，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への侵入を試みる通信やネットワーク内の不審な通信を検出してきた．しかしながら，近年問題となっている標的型攻撃は，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の個人を狙った攻撃であり，Firewallを通過するように設定されているプロトコルを利

[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 Prifecture University
[a)] mbb04025@edu.osakafu-u.ac.jp
[b)] aoki@kis.osakafu-u.ac.jp
[c)] aki@kis.osakafu-u.ac.jp

用して，RAT（Remote Access Trojan）と呼ばれる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，送り込まれたマルウェアを組織の人物が誤って実行することによって，端末がマルウェアに感染する．一度組織内の端末がマルウェアに感染すると，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搾取したい情報を探し出し，情報を搾取する．そのため，攻撃者による組織内ネットワークへの侵入後の不審な動きを速やかに検出するシステムが必要となっている．

本稿では，感染前の端末の通信ログを学習し，組織内の端末が標的型攻撃によってマルウェアに感染した時に，マルウェア感染端末が行う通信と，感染前の端末の通信の差異に着目することで，マルウェア感染端末を検出する手法を提案する．以下，第 2 節では関連研究について述べる．第 3 節では，マルウェア感染端末を検出する手法について説明する．第 4 節では本手法に対する予備実験と本手法の実験の結果に対する考察を述べ，第 5 節でまとめと今後の課題を示す．

2. 関連研究

一般社団法人 JPCERT は標的型攻撃の一連の流れを 7 つの段階に区分している [1, 2]．各段階の攻撃区分を表 1 に示す．偵察の段階で，攻撃者はインターネットから対象組織と組織内の人物に関する情報を取得し攻撃者を決定する．武器化の段階で，対象組織のネットワーク環境に存在する脆弱性を突いて遠隔操作を行うことができる RAT を作成する．デリバリの段階で，RAT を不正にインストールさせるマルウェアを添付したなりすましメールを対象組織の人物に送る．その人物がメールに添付されたマルウェアを実行した際に，エクスプロイトの段階に進み，予め作成した組織の端末の脆弱性を突くマルウェアの実行を試みる．エクスプロイトの実行に成功すると，インストールの段階に進み，端末に RAT を不正にインストールする．マルウェア感染後は C&C の段階で，攻撃者が C&C サーバと呼ばれるマルウェア感染端末に指示を行うサーバを用いて，感染端末に新たなマルウェアをダウンロードし，組織内の感染拡大を行う．そして C&C サーバから組織の内部情報の探索を試みる．そして最後に，探索で得られた内部データを加工（圧縮や暗号化）し，情報を持ち出す．以上が標的型攻撃の流れの手順である．

神谷ら [3] は，マルウェアを動的解析した結果から Firewall ログでの悪性リストを作成し，トラフィックログとのマッチングを行うことで，マルウェアに感染した端末を検出している．しかしながら，この手法での悪性リストは，シグネチャベースとしたアンチウイルスソフトウェアから作成されているため，パターンマッチングを回避するようにカスタマイズされている RAT や未知のマルウェアに対しての悪性リストが作成できないという課題がある．

表 1　標的型攻撃の攻撃段階

	攻撃の段階	概要
1	偵察	標的組織の人物の調査
2	武器化	脆弱性を突くマルウェアの作成
3	デリバリ	標的型メールの送信
4	エクスプロイト	脆弱性を突くコードの実行
5	インストール	RAT の不正インストール
6	C&C	組織内ネットワークの調査
7	目的の実行	情報窃取

三村ら [4] は，Proxy サーバの既知の正常な通信ログと不正な通信ログから，自然言語処理技術の一つである Doc2vec によってパラグラフベクトルを抽出し，そのベクトルを教師あり学習している．そして，検査用の Proxy サーバのログを評価することで，不正な通信ログを検出する手法を提案している．この手法では複数件のログを一つの文章として用いた際に，高精度に不正通信を検出できたことが示されている．

三浦ら [5] 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．この手法では悪性マクロが施されている MS 文書が入手困難であるという背景から，既知の良性マクロの特徴のみを自然言語処理技術の一つである LSI（Latent Semantic Indexing）を用いてベクトル化し，そのベクトルを教師なし学習させることで分類器を作成している．そして検査に用いる MS 文書を，学習させた分類器で評価することによって，悪性マクロを検出している．

Mao ら [6] は，Firewall ログに記載されている送信元 IP，宛先 IP，タイムスタンプで作成した 3 階のテンソルからテンソル解析することによって，正常トラフィックと類似しない端末を，マルウェア感染端末として検出している．

本稿では，正常な通信とマルウェアが行う通信との差異に着目する．そして，感染前の端末の通信ログを学習し，学習した感染前の端末の通信ログと類似しない端末を，マルウェア感染端末として検出する手法について提案する．本手法では，攻撃者によって操作される C&C の段階の感染端末の通信ログを，Firewall ログと Proxy サーバのログから検出する．

3. 提案手法

本手法は，組織内に設置されている端末が生成する通信ログから，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．しかしながら，標的型攻撃は攻撃対象によって攻撃手法が異なると考えられるため，予め攻撃手法を学習しておくことは難しい．そこで本稿では，端末が生成している正常な通信ログのみを学習することによって，マルウェア感染端末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．本手法は対象端末の通信ログの

文書からコーパスを作成し，自然言語処理技術の一つである Doc2vec でコーパスを学習することで，通信ログの文書をパラグラフベクトルとして表現する．次に，変換したパラグラフベクトルを One-Class SVM で学習することで，正常なベクトルの識別境界を決定する．そして，検査に用いる文書内文字列を Doc2vec に入力することで検査用文書をパラグラフベクトルとして表現し，そのベクトルを学習した One-Class SVM に入力することで，異常な通信をした端末を検出する．以下に詳細を述べる．

3.1 学習処理

3.1.1 通信ログでのコーパスの作成

通信ログの文書を Doc2vec を用いてベクトル表現するには，通信ログを単語毎に区切って記述したコーパスを用意する必要がある．本節では，通信ログからコーパスを作成する手順について述べる．通信ログの分割の流れを図 2 に示す．まず，過去の Proxy サーバのログを 1 日毎に分割し，1 日分の Proxy サーバのログから対象端末の IP アドレスが記載されている通信ログを抽出し，対象端末が 1 日に生成したログの文書を作成する．次に，対象端末が生成した 1 日のログから単語を抽出する．単語抽出の流れを図 3 に示す．対象端末の文書内に記載されている通信ログを区切り文字で分割し，意味のあるフィールドを抽出する．Proxy サーバのログで抽出する単語のフィールドは，HTTP ステータス，スキーム，プロトコル，メソッド，コンテンツタイプ，リクエスト URL，宛先ポート，要求ファイルの拡張子，クライアントが使用する User Agent の計 9 種類のフィールドとする．また，リクエスト URL は Proxy サーバのログの中でも一番特徴のあるフィールドである．本手法では，標的型攻撃の C&C の段階で攻撃者が行う，C&C サーバと感染端末との定期的な通信を検出するために，リクエスト URL から FQDN を抽出し，FQDN をドット（.）で更に分割することでリクエスト URL を細分化する．以上の処理を，学習期間中の全ての日の Proxy サーバのログに対して行い，対象端末が生成した学習期間中の各日の文書の単語を抽出し，日ごとの文書に文書 ID をつけることでコーパスを作成する．

3.1.2 文書のベクトル表現

3.1.1 節で作成したコーパスを Doc2vec で学習することによって，各日の文書をそれぞれパラグラフベクトルとして表現する．Doc2vec は Le [7] らによって提案されたモデルであり，Word2vec [8] の概念を文章全体に拡張したものとなっている．Word2vec は，中間層が 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，単語を文脈を考慮して学習し，単語をベクトルとして表現することができる．一方，Doc2vec は文章全体をベクトルとして表現することができるモデルである．Doc2vec を用いて文書内の文字列をパラグラフベクトルとして表現することで，文書同士の類似度を求めたり，文書のベクトルを推定することができる．本手法では Doc2vec の学習アルゴリズムの一つである DBoW（Distributed Bag-of-words）モデルによる学習を行う．DBoW モデルを図 4 に示す．ここで，d_i は文書 ID を表現する One-hot ベクトル，D はパラグラフベクトルの重み行列，W は単語ベクトルの重み行列，w_t は t 番目の単語が 1 でその他の要素が 0 の One-hot ベクトルを表す．DBoW モデルは，文書の文書 ID を表現する

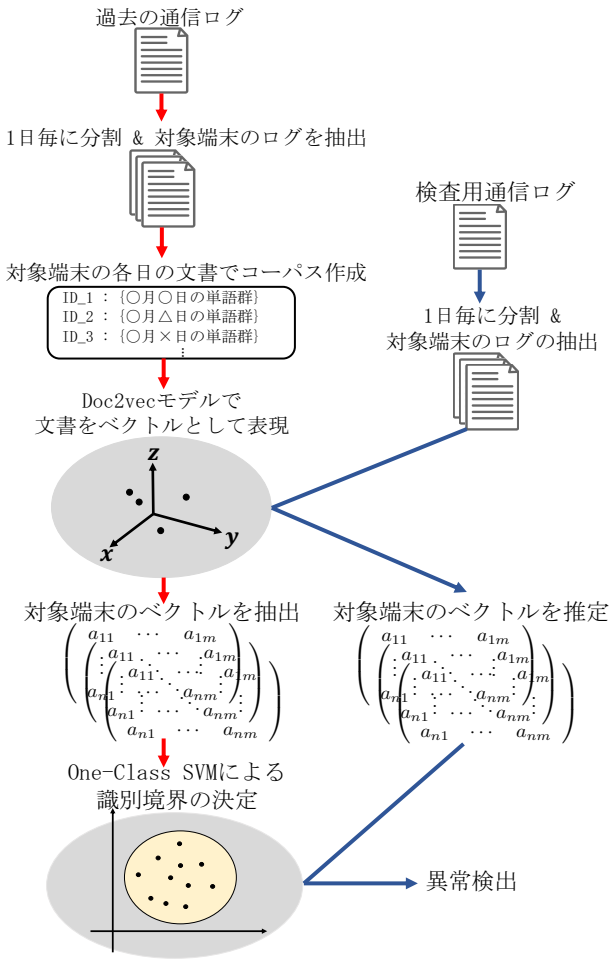

図 1 提案手法の概要

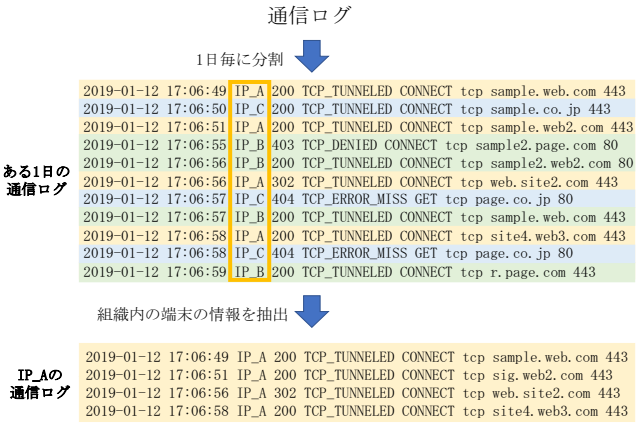

図 2 通信ログ分割の概要

Proxyサーバのログ

```
2019-01-12T17:05:35 src_IP - connect_method
_denied -  403 TCP_DENIED CONNECT - tcp
sample.web.com 5223 / - “ApplePushService/4.0
CFNetwork/760.9 Darwin/15.6.0 (x86_64) …
```

区切り文字で分割して
意味のある単語を抽出

```
Connect_method_denied 403 TCP_DENIED CONNECT - tcp
sample.web.com 5223 “ApplePushService/4.0
CFNetwork/760.9 Darwin/15.6.0 (x86_64) …
```

ドメインをドット
(.) で更に分割

```
Connect_method_denied 403 TCP_DENIED CONNECT - tcp
sample web com 5223 “ApplePushService/4.0
CFNetwork/760.9 Darwin/15.6.0 (x86_64) …
```

Firewallログ

```
2019-01-12T09:20:32, IP_A, IP_B, nat_src_IP, nat_dst_IP,
drop, ntp, 1001_WAN, WAN, 34511168, 1, syslog_server, 23373, 80,
0x19, tcp, allow, 400, 70, 330, JP, CA, any, …
```

区切り文字で分割して
意味のある単語を抽出

```
IP_B drop WAN ntp syslog_server tcp allow any JP CA any …
```

図 3　通信ログの単語抽出の概要

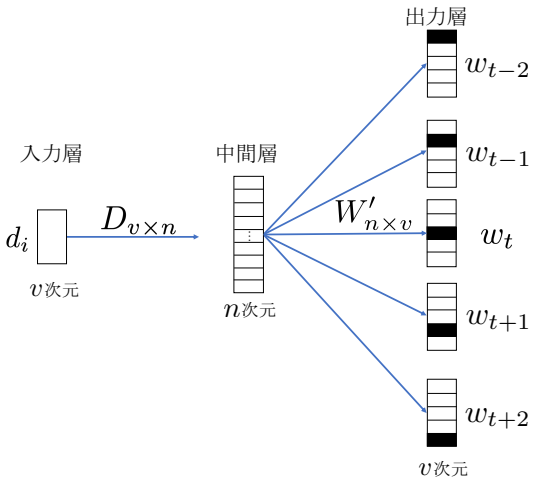

図 4　DBoW モデル

One-hot ベクトルを入力層に入力した時に，出力層に文書 ID の文書に含まれる単語の One-hot ベクトルが出現するように，文書の重み行列 D を更新するモデルとなっている．ここで，中間層の n 次元ベクトルがパラグラフベクトルとして表現される．本手法では，3.1.1 節で作成した，対象端末のコーパスに含まれる全ての文書 ID と，文書に含まれる単語の One-hot ベクトルを Doc2vec の DBoW モデルで学習する．そして，学習を終えた Doc2vec モデルに学習に用いた文書 ID を入力して得られる，中間層の n 次元ベクトルをパラグラフベクトルとして抽出する．

3.1.3　One-Class SVM による異常検出器の作成

3.1.2 節で抽出した各日のパラグラフベクトルを用いて異常検出器を作成する．本手法の異常検出器には One-Class SVM を用いる．One-Class SVM とは SVM (Support Vector Machine) アルゴリズムを応用した外れ値検知手法であり，SVM は 2 クラス分類の教師あり学習アルゴリズムであることに対して，One-Class SVM は単一のクラスのみを学習する教師なし学習アルゴリズムである．本手法では，3.1.2 節で抽出した各日のパラグラフベクトルの各次元の平均が 0 で分散が 1 となるように標準化し，標準化したパラグラフベクトルが分布するクラスを，One-Class SVM を用いて学習し，対象端末の異常検出器を作成する．以上の処理を，Proxy サーバのログに出現した全ての端末に対して同様の処理を行い，端末毎に異常検出器を作成する．

Firewall ログでも同様の処理を行い，端末毎に異常検出器を作成する．ただし，Firewall ログでコーパスを作成する際に抽出する単語のフィールドは，コンテンツタイプ，送信元 IP，宛先 IP，通過するパケットのセッション情報，宛先ポート，送信元エリア，宛先エリア，Firewall が判断した通過の可否，送信元国，宛先国，ポリシーの詳細の計 11 種類とする．

3.2　異常な端末の候補の抽出

3.1.3 節で生成した異常検出器を用いて，端末の異常判定を行う．まず，検査に用いる Proxy サーバのログを 1 日毎に分割し，分割した 1 日の Proxy サーバのログから組織内の IP アドレスが記載されている対象端末の通信ログを抽出し，対象端末が 1 日に生成したログの文書を作成する．そして，3.1.1 節と同様に，対象端末が 1 日に生成した通信ログから意味のある単語を抽出する．次に，抽出した単語群を 3.1.2 節で作成した対象端末の学習済みの Doc2vec モデルに入力することにより，対象端末の検査用の文書をパラグラフベクトルとして表現する．そして，検査用のパラグラフベクトルを，3.1.3 節で標準化した対象端末の平均と分散で標準化する．最後に，標準化された 検査用のパラグラフベクトルを 3.1.3 節で作成した対象端末の異常検出器に入力して異常を検出する．異常と判定された場合，マルウェア感染の疑いのある異常な端末の候補とする．以上の処理を，検査用のログに出現した全ての端末に対して同様の処理を行い，各々のパラグラフベクトルを対応する異常検出器に入力することで，マルウェア感染の疑いのある異常な端末の候補を抽出する．

Firewall の検査用ログでも同様の処理を行うことで，異常検出を行う．

3.3　マルウェア感染端末の判定

標的型攻撃の C&C の段階では，攻撃者はマルウェア感染端末を踏み台にして，組織内ネットワークの調査を行う．そのような調査が行われた痕跡は一般社団法人 JPCERT の報告 [1,2] によると，C&C サーバとマルウェア感染端

末との定期的な通信が Proxy サーバのログに，マルウェア感染端末による組織内ネットワークの調査が Firewall ログに，それぞれ残ることが示されている．そして C&C の段階での攻撃者による組織内ネットワークの調査は，長期間にわたって行われるため，攻撃者が定期的にマルウェア感染端末を用いて行った組織内ネットワークの調査の痕跡は，定期的に通信ログに記録されることが予想される．

そこで本稿でのマルウェア感染端末の判定は，3.1.3 節で作成された Proxy サーバ用の異常検出器と，Firewall ログ用の異常検出器に，端末のそれぞれの検査用のパラグラフベクトルを入力し，2 つの異常検出器が共に数日連続で不審な端末の候補と判定した場合にマルウェア感染端末として識別する．

4. 実験

4.1 各種ログでの精度評価に関する予備実験

本手法の有効性を確認するために，まず Proxy サーバのログと Firewall ログ各々でのマルウェア感染端末の検出精度を確認した．評価方法は，真陽性（True Positive rate, TP）と 偽陽性（False Positive rate, FP）による ROC（Receiver Operating Characteristic）曲線を用いた．ここで真陽性とは，異常なデータを正しく異常と判定した割合を示し，偽陽性とは，正常なデータを誤って異常データと判定した割合を示す．そして，ROC 曲線は上記の 2 つの割合の軌跡を表す．軌跡は真陽性の割合が高く且つ，偽陽性の割合が低いほど検出精度が高いことを示す．

4.1.1 予備実験の条件

Proxy サーバのログは 2018 年 10 月 1 日〜2018 年 11 月 29 日の，大阪府立大学の事務端末の通信ログで異常検出器を作成し，2018 年 11 月 30 日のログを検査用データとして学習済みの異常検出器に入力して評価した．Firewall ログは 2018 年 5 月 1 日〜2018 年 10 月 27 日の，大阪府立大学の事務端末の通信ログで異常検出器を作成し，2018 年 10 月 28 日のログを検査用データとして学習済みの異常検出器に入力して評価した．ROC 曲線による評価は，以下の 3 点で確認した．

(1) マルウェアに感染した端末の不審な通信ログが，感染前の通信ログと類似している場合の検出精度を確認するために，他の事務端末の通信ログをマルウェア感染端末の通信ログとみなして実験した．具体的には，事務端末の異常検出器に，同事務端末のパラグラフベクトルと，他の事務端末 20 台のパラグラフベクトルを入力した時の，TP と FP の割合を評価して ROC 曲線で描画した．

(2) マルウェアに感染した端末の不審な通信ログが，感染前の通信ログと類似しない場合の検出精度を確認するために，学生の端末の通信ログをマルウェア感染端末の通信ログとみなして実験した．具体的には，事務端末の異常検出器に，同事務端末のパラグラフベクトルと，学生の端末 20 台のパラグラフベクトルを入力した時の，TP と FP の割合を評価して ROC 曲線で描画した．

(3) マルウェアに感染した端末の不審な通信ログの類似性が，上記 2 つの条件の間の割合である場合の検出精度を確認するために，他の事務端末の通信ログに学生の端末の通信ログを挿入して，その通信をマルウェア感染端末の通信ログとみなして実験した．具体的には，事務端末の異常検出器に，同事務端末のパラグラフベクトルと，他の事務端末 20 台の通信ログに，学生の端末の通信ログを挿入することで生成されたパラグラフベクトルを入力した時の，TP と FP の割合を評価して ROC 曲線で描画した．

これら 3 点での Proxy サーバのログと Firewall ログによる異常検出器の評価結果をそれぞれ図 5，図 6 に示す．

4.1.2 予備実験結果及び考察

紫線は（1）の条件での ROC 曲線を，水色線は（2）の条件での ROC 曲線を，緑線は（3）の条件での ROC 曲線を示す．

Proxy サーバのログにおいて，紫線（1）の ROC 曲線は，マルウェアの通信を正しく検出できた TP の割合が全体的に低い結果となった．これは，職員同士はそれぞれ目的が似た用途で事務端末を使用するため，通信内容も類似したと考えられる．よって，マルウェアの Proxy サーバのログが，正常なログと類似している場合，異常な通信と正しく判定できる確率が低くなると予想される．次に，水色線（2）の ROC 曲線は，マルウェアの通信を正しく検出できた TP は全体的に高かったが，正常な通信をマルウェアの通信と判定した FP が少し高い曲線を描いた．FP の高さは，事務端末を使用する職員が毎日定期的に閲覧する Web サイトよりも，毎日違った Web サイトを閲覧することの方が多いことが原因であると予想される．しかしながら，TP が全体的に高い傾向が見られることから，学生と職員がそれぞれ違った用途で端末を使用していることを，Proxy サーバのログから判定できていると考えられる．よって，マルウェアの Proxy サーバのログが，正常なログと類似しない場合，FP が少し高めになるが，異常な通信と正しく判定できる確率が高くなると予想される．そして，緑線（3）の ROC 曲線は，紫線（1）と水色線（2）の間に描く結果となった．端末が標的型攻撃によってマルウェアに感染した場合，その端末から生成されるマルウェアの Proxy サーバのログが，正常なログと類似しなければしないほど，異常な通信と正しく判定できる確率が高くなると予想される．

Firewall ログにおいて，紫線（1）の ROC 曲線は，Proxy サーバのログでの ROC 曲線と類似した傾向が見られた．これは，Proxy サーバのログと同様の理由が考えられる．マルウェアの Firewall ログが正常なログと類似している場

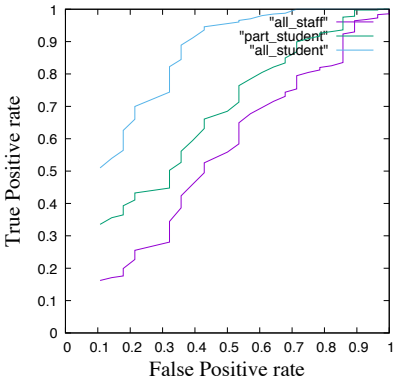

図 5　Proxy サーバのログの ROC 曲線.

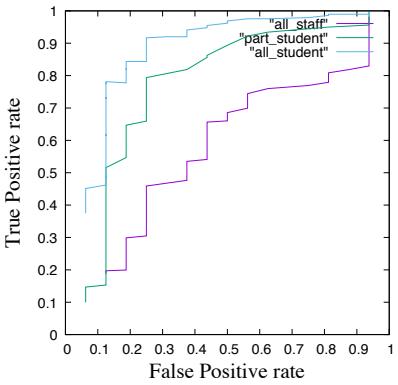

図 6　Firewall ログの ROC 曲線.

合，異常な通信と正しく判定できる確率が低くなると予想される．次に，水色線（2）の ROC 曲線は，Proxy サーバのログと比べると TP が全体的に高く，FP が全体的に低い結果となった．これも Proxy サーバのログと同様，学生と職員はそれぞれ違った用途で端末を使用していることの差が，パラグラフベクトルに現れたからであると考えられる．マルウェアの Firewall ログが正常なログと類似しない場合，異常な通信と正しく判定できる確率が高くなると予想される．そして，緑線（3）の ROC 曲線は，紫線（1）と水色線（2）の間に描く結果となった．このことから，Proxy サーバのログと同様，端末が標的型攻撃によってマルウェアに感染した場合，その端末から生成されるマルウェアの通信が正常な通信と類似しなければしないほど，異常な通信を正しく判定できる確率が高くなると予想される．

4.2　C&C の段階でのマルウェア感染端末の検出実験

4.2.1　実験条件

実験では，Proxy サーバのログは 2018 年 10 月 1 日～2018 年 12 月 25 日の，大阪府立大学の事務端末の通信ログで異常検出器を作成し，2018 年 12 月 26 日～2018 年 12 月 28 日の検査用のログに，マルウェアの挙動を模して作成したログを挿入した通信ログと，何も挿入していない正常な通信ログを，学習済みの異常検出器に入力することで評価した．Firewall ログは 2018 年 5 月 1 日～2018 年 12 月 25 日の，大阪府立大学の事務端末の通信ログで異常検出器を作成し，2018 年 12 月 26 日～2018 年 12 月 28 日の検査用のログに，マルウェアの挙動を模して作成したログを挿入した通信ログと，何も挿入していない正常な通信ログを，学習済みの異常検出器に入力することで評価した．また異常判定は，Proxy サーバのログと Firewall ログで生成した 2 つの異常検出器が，共に 2 日以上続けて異常と判定した端末をマルウェア感染端末として判定した．ここで，マルウェアの挙動を模して作成したログは，攻撃者が C&C サーバを経由して感染端末に定期的にアクセスし，感染端末を使用して組織内のサーバにスキャンを試みたり，感染端末が許可されていない別セグメントへのアクセスを試みた時の，Proxy サーバのログと Firewall ログである．実験は，大阪府立大学の 102 台の事務端末の通信ログで行った．Proxy サーバのログで異常な端末と判定した端末数の結果を表 2 に示す．また，Firewall ログで同様の判定を行った結果を，表 3 に示す．そして，2 つの異常検出器から判定したマルウェア感染端末の判定結果を表 4 に示す．

4.2.2　実験結果と考察

One-Class SVM で異常と判定する閾値を，Proxy サーバのログでは 4.1.2 節での図 5 より，FP が全体的に高くなるが，TP を全体的に高く保てた時の異常検出器のハイパーパラメータ γ である 0.0116 に設定し，Firewall ログは 4.1.2 節での図 6 より，Proxy サーバのログと同様に TP を高く保てた異常検出器のハイパーパラメータ γ である 0.0011 に設定して実験した．

表 2 より，感染端末の Proxy サーバの通信によって検査期間中に 2 日以上続けて異常な通信と判定された事務端末は 102 台中 90 台で，残りの 12 台を正常な事務端末と誤検知した．また，正常通信で同様の実験を行った結果，正常な通信で検査期間中に 2 日以上続けて異常と誤検知した事務端末は 102 台中 63 台で，残りの 39 台の事務端末を正常な端末と正しく検出できた．Proxy サーバで正常な通信による 63 台の事務端末の誤検知は，異常検出器のパラメータで FP が高くなるように閾値を設定したため，正常な通信を異常な通信と誤検知した割合が多くなったと考えられる．また，異常な通信を異常と検出できなかった 12 台の事務端末の検査に用いた文書を調査すると，それらの文書は全て，正しく異常検出できていた 90 台の端末の検査用の文書と比べると，10 倍近く多い分量のログが記録されていたことがわかった．そのため，誤検知した 12 台の端末の通信ログは，マルウェアの通信ログが正常な通信ログに紛れた可能性が考えられる．Doc2vec は文書をパラグラフベクトルとして表現するときに，学習時のコーパスに存在していた既知の単語の情報のみでベクトルを推定する．そのため，異常な通信に含まれる未知の単語の量が既知の単語に比べて著しく少ない場合，正常な通信のパラグラフベクトルに類似したパラグラフベクトルになるため，異常な

表 2　Proxy サーバのログでの感染端末の検出数

	正常通信		感染端末の通信	
	正常	異常	正常	異常
Proxy ログ	39/102	63/102	12/102	90/102

表 3　Firewall ログでの感染端末の検出数

	正常通信		感染端末の通信	
	正常	異常	正常	異常
Firewall ログ	64/102	38/102	3/102	99/102

表 4　二つの通信ログによる異常判定結果

正常通信		感染端末の通信	
正常	異常	正常	異常
81/102	21/102	14/102	88/102

通信を検出できなかったと考えられる．

表 3 より，感染端末の Firewall ログによって検査期間中に 2 日以上続けて異常な通信と判定された事務端末は 102 台中 99 台で，残りの 3 台の事務端末を正常な端末と誤検知した．また，正常な通信で同様の実験を行った結果，正常な通信で検査期間中に 2 日以上続けて異常な端末と誤検知した端末数は 102 台中 38 台で，残りの 64 台の事務端末を正常な端末と正しく検出できた．正常な通信を異常と判定した 38 台の事務端末の Doc2vec での学習時に用いた文書を調べると，各日の文書のログの分量が極端に少なかったことがわかった．Doc2vec での学習には，文書内の単語の量によって学習率を調整する必要がある．しかしながら本実験では，各端末が一定の量の通信ログを生成している想定で実験したため，全ての端末に対して同一の学習率を設定した．そのため，ログの分量が極端に少ない文書に対して，過学習を起こしていた可能性が考えられる．この問題については，文書の分量に合わせて学習率を調整することで解決できると考えられる．

表 4 より，二つの通信ログから異常判定を行った結果は全体的に見ると，異常な端末を異常と判定できた数が多いが，正常な通信を異常と誤検知した端末数が 21 台で少し多い結果となった．Proxy サーバのログでは正常なログの量が多く，マルウェアの通信ログが紛れたために検出できなかったが，検査用のログを一定量毎に分割するなどの方法で，マルウェアの通信の検出漏れを解決できると考えられる．Firewall ログでは，ログの分量に応じて学習率を調整すれば，正常な通信を異常と誤検知する確率を減らすことができると考えられる．

5. まとめ

本稿では，Proxy サーバのログと Firewall ログに Doc2vec を適用して，感染前の端末が生成している一日分の通信ログをパラグラフベクトルで表現した．そして，それらのベクトルが分布しているクラスを One-Class SVM で学習することで，マルウェア感染の疑いのある端末を検出する手法を提案した．本手法では，Proxy サーバのログと Firewall ログを組み合わせることによって，高精度にマルウェア感染端末を検出できた．しかしながらマルウェアの通信を高精度に検出する閾値では，正常な通信を異常な通信と誤検知する確率が高くなった．今後の課題として，分量の少ない文書での過学習を回避するために，ログの文書の分量に応じた学習率の調整と，マルウェアの通信が正常な通信ログに紛れないための，文書の一定量の分割などが挙げられる．

参考文献

[1] JPCERT コーディネーションセンター：ログを活用した高度サイバー攻撃の早期発見と分析 (online)，入手先 <https://www.jpcert.or.jp/research/apt-loganalysis.html>(2016.10.19)．
[2] JPCERT コーディネーションセンター：高度サイバー攻撃への対処におけるログの活用と分析方法 1.1 版 (online)，入手先 <https://www.jpcert.or.jp/research/apt-loganalysis.html>(2016.10.19)．
[3] 神谷和憲，青木一史，中田健介，佐藤達，倉上弘，谷川真樹：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，Vol2015，No.1 pp433-434(2015)．
[4] 三村守，田中秀磨：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，コンピュータセキュリティシンポジウム 2017 講演論文集，3B4-4，pp1418-1425(2017)．
[5] 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 2018 講演論文集，2C3-2，pp462-469(2018)．
[6] Mao, H. H., Wu, C. J., Papalexakis, E. E., Faloutsos, C., Lee, K. C., and Kao, T. C.:MalSpot: Multi 2 malicious network behavior patterns analysis, Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Cham(2014).
[7] Le, Q., and Mikolov, T.:Distributed representations of sentences and documents, International Conference on Machine Learning, pp1188-1196(2014).
[8] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J.:Distributed representations of words and phrases and their compositionality, Advances in neural information processing systems, pp3111-3119(2013).