情報処理学会論文誌 Vol.62 No.3 838–847 (Mar. 2021)

アプリケーション識別機能付き ファイアウォールのログを対象とした 機械学習による自己らしい通信の識別手法

市之瀬 樹生[1,†1] 佐藤 聡[2,3,a)] 新城 靖[3] 三宮 秀次[2,3] 星野 厚[1,4]

受付日 2020年6月22日, 採録日 2020年12月1日

概要：コンピュータやネットワークの脅威の一部では，ふだんと異なる，自己らしくない通信が発生する．ある一連の通信が自己による通信に似ているかを識別する手法は，役立つと考えられる．本研究は，アプリケーション識別機能を有するファイアウォールのログを用いて，識別対象者の通信の振舞いを学習し，ある通信が自己らしい通信であるか否かを識別する手法を提案する．提案手法では，学習フェーズにて，入力をファイアウォールログから生成した通信アプリケーション列とし，出力を識別対象者らしさを表す数値とする識別器を作成する．認識フェーズでは，学習フェーズで作成した識別対象者専用の識別器に対して，識別したい通信のファイアウォールログに学習フェーズと同じ手法を適用して生成した通信アプリケーション列を入力して識別を行う．提案手法に基づいた実験では，AUC の平均値は 0.76037 となり，識別能力があるという結果となった．また，1 カ月の通信ログを対象にした個人ごとの識別器の生成時間は約 1 時間であった．これらより提案手法が有効であることを示した．

キーワード：ログ解析，機械学習，識別

Your Own Traffic Discrimination Method by Machine Learning Using Log of Firewall with Application Identification Function

Tatsuki Ichinose[1,†1] Akira Sato[2,3,a)] Yasushi Shinjo[3] Shuji Sannomiya[2,3] Atsushi Hoshino[1,4]

Received: June 22, 2020, Accepted: December 1, 2020

Abstract: Some of the threats of computers and networks cause unusual traffic unlike your own. We believe that a discrimination method to determine whether a certain series of traffic is similar to your own traffic or not is useful. In this paper, we propose such discrimination method by learning one's behavior of the traffic based on the log of the firewall with application identification function. For the learning phase of the proposed method, we create a classifier that receives a communication application sequence generated from the firewall log and outputs a numerical value that represents how much the sequence is like target user's. In the recognition phase, we discriminate the communication application sequence generated by applying the same method as in the learning phase to the firewall log using the classifier. In the experiments of the proposed method, the average value of AUC was 0.76037, which means that the proposed method is considered to achieve an acceptable discrimination ability. Moreover, it took about one hour to generate the classifier by using firewall logs for one month for each individual who is targeted. These results show that the proposed method is effective.

Keywords: log analysis, machine learning, discrimination

1. はじめに

コンピュータを利用するにあたってセキュリティ対策は

[1] 筑波大学システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305–0006, Japan
[2] 筑波大学学術情報メディアセンター
Academic Computing and Communications Center, University of Tsukuba, Tsukuba, Ibaraki 305–0006, Japan
[3] 筑波大学システム情報系
Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Ibaraki 305–0006, Japan
[4] 株式会社チノウ
Chinoh, Inc, Tsukuba, Ibaraki 305–0844, Japan
[†1] 現在，ソフトバンク株式会社
Presently with SoftBank Corp.
[a)] akira@cc.tsukuba.ac.jp

必須である．しかし，近年は標的型攻撃をはじめとする高度サイバー攻撃が増加しており，既知の攻撃・脅威に対処する従来の手法だけでは不十分になっている．そのため，攻撃自体を防ぐのはもちろんだが，攻撃を受けることを前提に，コンピュータに仕組まれた脅威をいち早く検出する技術が，重要視されている [10]．

この脅威の一部は，ふだんと異なった自己らしくない通信を行う．具体的には，マルウェアの感染拡大，C2 サーバ（Command & Control Server）との通信，情報の盗み出しや，コンピュータを乗っ取った侵入者が行った通信などである．このように，ふだんと異なった自己らしくない通信が行われていることを識別することができれば，コンピュータやネットワーク内に何らかの脅威が存在する可能性を示すことに役に立つと考えられる．

本研究は，アプリケーション識別機能を有するファイアウォールのログの中から識別対象者の通信のログだけを用いて，識別対象者の通信の振舞いを学習し，ある通信が自己らしい通信であるか否かを識別する手法を提案する．

本提案手法の利点は主に 3 点存在する．1 つ目がネットワークトラフィックのみから識別を行えることである．このことは，プロセッサやメモリなどの資源制約があるなどの理由によりアンチウィルスソフトウェアを実行できない機器がある環境などに容易に導入できることを意味する．2 つ目がマルウェア以外の脅威の識別も行えることである．従来手法ではマルウェアによる通信の特徴を学習し識別するため，乗っ取りなどによって行われるマルウェアではない通信を識別することはできなかった．しかし，本提案手法では自己らしいか否かを識別する．そのため，マルウェアによる通信はもちろん他人による通信も，ふだんと異なる通信であるとして識別することが可能である．3 つ目が識別対象者の通信のログだけを用いて学習を行うことである．このことは，マルウェアによる通信の記録を収集する必要がないことや，通信の記録が収集されていない新しいマルウェアによる通信を識別できることを意味している．

一方で，本提案手法では，個人ごとに識別器を作成するため，従来の手法と比較して多くの識別器を作成する必要がある．このことから識別機を作成する学習時間を削減するため，処理するデータを簡略化する必要がある．本提案手法では，その簡略化した通信データとして，アプリケーション識別機能を有するファイアウォールのログを用いている．

本提案手法では，学習フェーズで，入力をファイアウォールログから生成した通信アプリケーション列とし，出力を識別対象者らしさを表す数値とする識別器を識別対象者ごとに作成する．認識フェーズでは，学習フェーズで作成した識別対象者専用の識別器に対して，識別したい通信のファイアウォールログに学習フェーズと同じ手法を適用して生成した通信アプリケーション列を入力して識別を行う．識別精度は識別対象者の識別器に識別対象者自身の通信アプリケーション列を入力した場合と，識別対象者ではない，他ユーザの通信アプリケーション列を入力した場合で，出力される識別対象者らしさに差がでるか，どの程度の差であるかをもとに評価する．また本研究では，ファイアウォールログで適用する処理の有無を変えて 4 種類のデータを生成し，識別精度と学習時間を計測する実験を行った．

2. 関連研究

2.1 通信情報を用いたマルウェア検知

通信情報列を用いてその通信がマルウェアによる通信か否かを，分類するマルウェア検知はすでに多く行われている [2], [14], [15], [17]．特に HTTPS トラフィックからも読み取れるタイムスタンプやホストアドレスの利用 [14] や，IP アドレスとポート番号の組合せや通信量を利用 [15] して，RNN/LSTM [6] を用いてマルウェアの検知を試みる研究もなされている．これらの研究はマルウェアの特徴のみを検出するため，未知の特徴を持つマルウェアや不正ログインによる乗っ取りなどを検出することは難しい．

また，TLS ハンドシェイク時に得ることのできる，暗号スイートや公開鍵の長さなどといった情報から分類を行おうとしている研究 [1], [2] や，トラフィックのペイロードのバイナリ分布をみる研究 [14], [17] がある．これらはパケット，または TLS メッセージ単位の詳細な解析が必要である．そのためログデータの保管領域や解析資源を多く必要とする．

マルウェアの動作が端末の通信に影響を及ぼすことを利用し，マルウェアの侵入を発見する手法が研究 [9], [11] されている．また，プロセスと通信の傾向に関連があることも示されている [7]．これらのことから端末がマルウェアに感染した場合にふだんと異なる通信を行うため，本提案手法を用いて通信アプリケーション列より自己らしい通信か否かを判定することにより，マルウェアの感染を識別することが可能である．

2.2 LSTM を用いたマルウェア本体の検知

RNN の特徴抽出や自然言語モデル，バイナリファイルから抽出した画像などを使用することで，あるファイルがマルウェアであるかどうかを識別する研究 [3], [12], [19] やアプリケーションの逆コンパイルを行い，そのソースコードを LSTM を用いて解析することでマルウェアを発見する研究 [20] がなされている．またプロセスログをもとにそのプロセスがマルウェアによるものかどうかを識別する研究 [16], [18] も存在する．しかし，これらはマルウェア本体を検知する技術である．

そのため，コンピュータの内部に検知器を設置しておく必要がある．

2.3 LSTM を用いた HAL（Human Activity Recognition）に関する研究

LSTM を用いて主にビデオやセンサの情報から，人間の行動を認識する研究が行われている．Zhao らの研究 [21] では，日常生活をウェアラブルセンサを着用した状態で行い，そこで収集した活動情報をもとに双方向 LSTM を用いて活動認識を行っている．Pouyanfar らの研究 [13] では，ビデオ画像から 2 層の双方向 LSTM を用いてビデオを分類している．Chen らの研究 [4] では，モバイルセンシングアプリケーションを用いた人間の活動の認識を行っており，LSTM を用いた特徴抽出を行っている．

これらの研究は，ある者の振舞いから個人を識別している．振舞いによる識別という範疇ではあるが，本提案手法はある者の通信の振舞いにより個人を識別している点で異なっている．

3. 提案手法

本研究は，アプリケーション識別機能を有するファイアウォールのログを用いて，識別対象者の通信の振舞いを学習し，ある通信が自己らしい通信であるか否かを識別することを目的とする．そのために，まず，学習フェーズにおいて，入力を識別対象者の通信に関するファイアウォールのログに処理を適用した通信データとし，出力を通信データの識別対象者らしさとする，識別対象者専用の識別器を作成する．識別フェーズでは，学習フェーズで作成した識別対象者専用の識別器に対して，識別したい通信ログに対して学習フェーズと同じ処理を適用した通信データを入力し，その通信が自己らしい通信であるか否かを表す数値を出力させる．本提案手法の概要を**図 1** に示す．

本提案手法の特徴は，識別器を識別対象者ごとに作成することと，学習データとして識別対象者本人の通信ログしか使わない点にある．

本研究ではユーザの特徴を学習することにより，端末の乗っ取りなど他ユーザによる通信の検出にも応用できる．学習においてマルウェアの通信データが不要である．

3.1 想定するネットワーク環境

本提案手法が想定するネットワーク構成は，**図 2** に示すような企業などでは一般的なネットワーク構成とする．この環境に求める要件を以下に示す．

- ユーザの端末からインターネットへの通信については，すべてファイアウォールを経由する．
- ファイアウォールはアプリケーション識別機能を有している．
- ユーザの端末とインターネット間の通信に関するファイアウォールの設定は，外向き通信は許可，内向き通信は拒否としている．
- ユーザの端末が利用する DNS フルサービスリゾルバは，ファイアウォールよりユーザの端末側に設置されている．
- ユーザの端末はどの時刻にどの IP アドレスを使っていたかを把握できている．

この要件を満たすネットワーク環境は，一般的な組織などのネットワーク環境とほぼ同じである．また，この要件を満たす場合，ファイアウォールで記録されるユーザ端末に関する通信はユーザの端末が要求した通信だけとなり，DNS 問合せによる通信は記録されない．

3.2 識別器の作成

本提案手法における識別器の概要を**図 3** に示す．識別器は Embedding 部，ニューラルネットワーク部，自己度計算部の 3 つの部分から構成される．通信アプリケーション列は，識別フェーズでの識別器入力データとして利用するだけでなく，識別器の Embedding 部の学習用データとして利用するとともに，学習フェーズの識別器入力データとなる教師データを作成するためにも利用する．

3.2.1 通信アプリケーション列

本提案手法での識別器の入力には，ファイアウォールの通信ログに記録されているアプリケーション名のみを用いる．ファイアウォールの通信ログから当該ユーザの ID と当該ユーザが用いていた IP アドレスが一致しているログを抽出し，発生時刻順に並んだアプリケーション名の列を作成する．その先頭から 30 のウインドウサイズで切り出す．その後その先頭から 1 つ次のアプリケーション名を先頭にして，同様に切り出す．これを順次繰り返すことにより複数の入力データを生成する．本研究ではこのようなウ

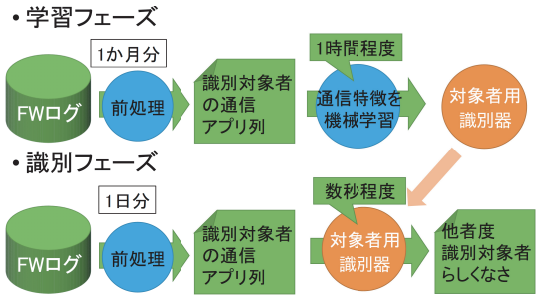

図 1 提案手法の概要
Fig. 1 Overview of the proposed method.

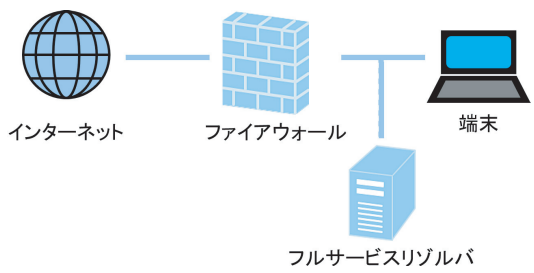

図 2 想定するネットワーク構成
Fig. 2 Assumed network environment.

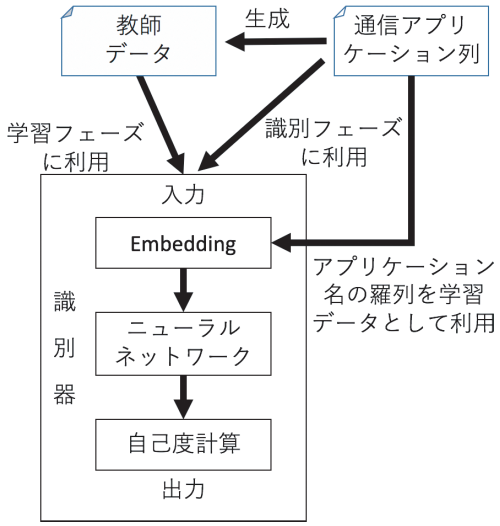

図 3 識別器の概要
Fig. 3 Overview of the classifier.

インドウサイズごとに区切られた通信アプリケーションのログを，**通信アプリケーション列**と呼ぶ．

なお，後述する 6 章で示した方法と同じ方法を用いて，ウインドウサイズを 10，20，30，40，50 と変化させて測定する予備実験を行った．その結果，最も良い性能はウインドウサイズが 30 のときに得られた．

また，実験に用いた環境では，DHCP のリリース時間が短く設定されており，長時間のネットワーク利用がない場合に，IP アドレスが変更になるという特徴がある．このため，同一ユーザ ID を持つ時間的に隣接するアプリケーション通信のログにおいて，IP アドレスが一致しない場合があり，その結果，1 つの通信アプリケーション列の中に長時間を空けて利用されたアプリケーションが含まれることはなかった．

3.2.2 教師データ

本提案手法で識別器を作成するため教師データとしては，識別対象者自身の通信アプリケーション列である自己データと，識別対象者ではない通信アプリケーション列である他者データが必要である．他者データは，識別対象者の通信ログで現れたアプリケーションをもとに，ウィンドウサイズと同じ 30 回分のアプリケーションをランダムに選択して通信アプリケーション列を生成することで作成した．

ただし，このときランダムに生成された通信アプリケーション列が自己データの中に存在した場合は再生成を行う．このようにすることで，他者データは，自己データに存在しない通信アプリケーション列とした．

3.2.3 Embedding 部

ニューラルネットワークに入力するために行う，アプリケーションのベクトル化である Embedding としては，Word2vec を用いる．Word2vec は，自然言語処理の分野でよく利用されている手法 [8] である．

本提案手法では，通信アプリケーション名の羅列を単語の羅列に見立て，自然言語の文章と同じ要領でアプリケーション名を 32 次元ベクトルに変換した．学習フェーズにおいて，出現頻度が 1 となるアプリケーション名は，まとめて unknown というアプリケーション名として学習を行った．また，識別フェーズにおいて，評価器の入力となる通信アプリケーション列で初めて出現したアプリケーション名も unknown として取り扱った．Word2vec の学習方法にはウインドウサイズを前後 1 単語とした skip-gram を用いる．また，これらの学習は，識別対象者ごとにバッチサイズを 128 としたミニバッチ学習で行う．

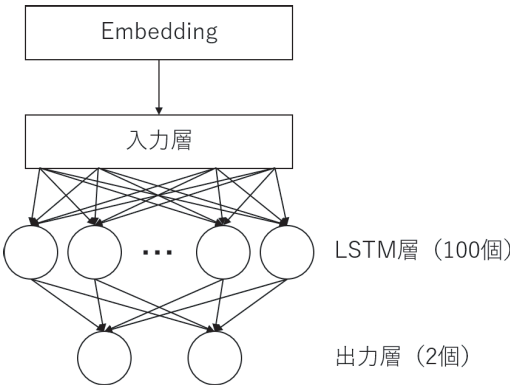

図 4 ニューラルネットワークのモデル
Fig. 4 Model of neural network.

なお，後述する 6 章で示した方法と同じ方法を用いて，Word2vec が変換する次元数を 8，16，32，64 と変化させて測定する予備実験を行った．その結果，最も良い性能は次元数が 32 のときに得られた．

3.2.4 ニューラルネットワーク部

本提案手法で用いるニューラルネットワークは RNN（Recurrent Neural Network）の中の LSTM（Long Short-Term Memory）を用いる．そのモデルを**図 4** に示す．RNN の構造は入力層，LSTM 層（1 層），出力層の三層構造とした．LSTM 層のセル数は 100 とした．

出力は 2 次元のベクトルとし，自己の通信を $[1,0]$，他者の通信を $[0,1]$ としてラベル付けし，学習を行った．

なお，後述する 6 章で示した方法と同じ方法を用いて，LSTM 層のセル数を 25，50，75，100，125，150 と変化させて測定する予備実験を行った．その結果，最も良い性能はセル数が 100 のときに得られた．

識別器のニューラルネットワークの部分は，ミニバッチにより学習を行う．この際のバッチサイズは 200 とした．ミニバッチは次の操作で作成する．最初に 0 から 1 の乱数を発生させる．その乱数が 0.1 未満であった場合は自己データからランダムに 1 つ選択し，それを Embedding 部でアプリケーション名をベクトルに変換した後にミニバッチに追加する．一方で乱数が 0.1 以上の場合は他者データを 1 つ生成し Embedding 部でアプリケーション名をベクトルに変換した後にミニバッチに追加する．これを 200 回

繰り返すことでサイズ 200 のミニバッチを作成する．このことから，おおよそ自己の通信データ 1 割，他者の通信データ 9 割の比率で，教師データを生成する．

なお，後述する 6 章で示した方法と同じ方法を用いて，自己の通信データに対する他者の通信データの割合を 1 倍，9 倍，99 倍と変化させて測定する予備実験を行った．その結果，最も良い性能は自己の通信データに対して他者の通信データの割合が 9 倍のときに得られた．

3.2.5 自己度計算部

本提案手法の識別器が出力する 2 次元ベクトル [a, b] から通信アプリケーション列がどれほど自己らしいかを表す値を計算する．本論文はこの値を**自己度**と呼称する．ここで，a は自己らしさを表す数値，b は他者らしさを表す数値である．これらの値はおおよそ 0 から 1 の値をとる．

自己度は，0 のときに「どちらともいえない」を意味し，正のときに識別対象者らしい通信，負のときは他者らしい通信であることを表すように次のように定義する．

$$\begin{cases} 0 & (a = b = 0) \\ \dfrac{a}{a+b} - 0.5 & (\text{otherwise}) \end{cases} \tag{1}$$

この自己度は，その絶対値が大きければ大きいほど確信度が高くなる．この自己度がある閾値を下回った場合に自己らしくない通信であるとして，アラートを出すことに応用できる．

4. 実装方法

本研究では著者らが所属する大学で運用されている無線 LAN システムを対象に実装を行った．このシステムは，3.1 節で述べた想定する環境の要件に一致している．

4.1 使用するファイアウォール

このシステムで使われているファイアウォールは Palo Alto Networks 社製のファイアウォール PA-7050（version 7.1）である．このファイアウォールは通過したパケットを解析し，その通信を行ったアプリケーションの名前を通信ログに付加する．このファイアウォールは SSL，web-browsing，google-base などのアプリケーションが識別可能であり，全 2,962 種類[*1]から決定される．ウェブブラウジングの内容などは，単に web-browsing だけでなく，google-base や twitter-base などのよく利用されるサービスも識別される．また，このファイアウォールでは独自のアプリケーションの定義が可能である．本大学ではイントラネットのサーバにアクセスする通信について，独自のアプリケーションを 2 種類定義している．

このファイアウォールは，通信終了したときに，アプリケーション名やアプリケーションによる通信が開始された時刻などを通信ログに出力するように設定してある．すなわち，ファイアウォールログはおおよそ通信終了時刻順に記録されている．

[*1] https://applipedia.paloaltonetworks.com/

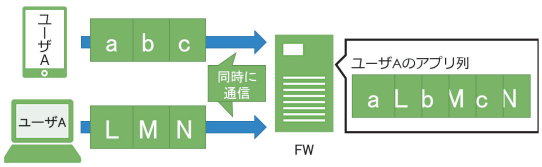

図 5 複数端末による通信
Fig. 5 Communication with multiple terminals.

実装対象となる無線 LAN システムでは，ユーザは接続する際に認証が必要となる．認証が成功すると，利用している IP アドレスとユーザ ID がファイアウォールに通知され，ファイアォールはその情報を保持し，出力する通信ログにユーザ ID を付加する．

4.2 通信アプリケーション列の生成

実装対象としたファイアウォールの通信ログの中から，以下に示す 4 項目を用いて通信アプリケーション列を作成した．

- ユーザ ID
- 送信元 IP アドレス
- 通信したアプリケーション
- 通信開始時刻

まずはじめに，ファイアウォールの通信ログからユーザのログを抽出する．これにはユーザ ID を用いた．しかし，**図 5** のようにユーザが同時刻に複数の端末を利用している場合には，異なる複数の端末の通信から通信アプリケーション列として生成される．そこでユーザ ID と送信元 IP アドレスの組合せで抽出を行った．生成された通信アプリケーション列はすべてユーザのものとして取り扱った．

次に，ファイアウォールの通信ログを通信開始時刻順に並べ替えた．これは，ユーザ自身の操作がまったく同じであっても，インターネット回線の速度などの外部の要因によって通信終了時刻が変わる可能性を排除するために行った．通信開始時刻順に並べ替えることにより，リアルタイムに順次識別を行うことができない，並べ替え処理がオーバヘッドになるという問題点がある．

最後に，通信ログからアプリケーション名だけを切り出すことにより，通信アプリケーション列を作成した．

4.3 実装環境

識別器は，TensorFlow の BasicLSTMCell を利用し，Python で開発した．また，実装した計算機は一般的なゲーミング用の PC を用いた．そのスペックを**表 1** に示す．

表 1 機械学習に用いた計算機のスペック
Table 1 The specifications of the computer used for machine learning.

項目	詳細
CPU	Intel Core i5-6400
GPU	NVIDIA GeForce GTX1050Ti
メインメモリ	8 GB

5. 評価方法

本研究はある通信が識別対象者によるものか否かを識別できることを目的としている．そこで，ある通信アプリケーション列が，識別対象者自身によるものか，識別対象者ではない別ユーザによるものかを，正確に識別することができるか否かで，識別器の評価をすることとした．具体的には，まず識別対象者の訓練用データを用いて対象者用の識別器を作成する．次に，その識別器に識別対象者自身の通信アプリケーション列を入力した場合と，識別対象者ではないユーザの通信アプリケーション列を入力した場合で，出力される自己度の違いに着目する．識別器に，識別対象者自身の通信アプリケーション列を入力した場合は，自己の通信であることから高い自己度が出力され，識別対象者ではない，他ユーザの通信アプリケーション列を入力した場合は，自己の通信ではないことから低い自己度が出力されることが期待される．この期待どおりの出力が得られるか否かで評価を行う．

5.1 評価用データ

性能評価で利用するファイアウォールログは，提供に同意したユーザ自身が，ファイアウォールを管理している学術情報メディアセンターから開示された各自のファイアウォールログを利用した．

今回は提供に同意をいただいた 5 人分の，ファイアウォールログを利用した．このログデータは 2019/4/1 から 2019/4/30 の間に記録されたものである．本実験で用いたファイアウォールログの件数を**表 2** に示す．

また，評価データを採取した時間帯に識別対象者らの被験者が利用していたデバイス（OS）の一覧を**表 3** に示す．

5.2 ROC 曲線と AUC

ROC（Receiver Operatorating Characteristic）曲線とは受信者動作特性曲線とも呼ばれ，機械学習の識別性能の良さや閾値を決める際に用いられる．具体的な ROC 曲線例を**図 6** に示す．ROC 曲線の横軸は偽陽性率（FPR：False Positive Rate），縦軸は真陽性率（TPR：True Positive Rate）を表す．本研究では横軸の偽陽性率が識別対象者の通信であるにもかかわらず誤って識別対象者でない通信であると判定した比率，縦軸の真陽性率が識別対象者でない通信を，正しく識別対象者でない通信であると判定した比率である．

ROC 曲線を作成するために必要な偽陽性率と真陽性率の算出方法を，**図 7** に示す．最初に識別対象者の訓練用の通信アプリケーション列を用いた学習により作成した識別器を準備する．この識別器に，識別対象者自身の評価用の複数の通信アプリケーション列を入力した場合に得られる複数の自己度と，他者の複数の通信アプリケーション列を入力した場合に得られる複数の自己度を取得する．識別対象者自身のデータを入力して得られた自己度は全体的に大きな値であり，一方で他者のデータを入力して得られた自

表 2 実験で用いたファイアウォールログの件数
Table 2 Number of firewall log used in the experiment.

項目	件数
識別対象者（訓練用）	41,237 件
識別対象者（評価用）	16,123 件
ユーザ A	19,044 件
ユーザ B	56,632 件
ユーザ C	9,374 件
ユーザ D	19,617 件

表 3 被験者が使ったデバイスと OS
Table 3 The devices (OS) used by the experimenters.

ユーザ	デバイス（OS）
識別対象者	ノート PC（Windows 10） スマートフォン（Android 8）
ユーザ A	ノート PC（Windows 10） スマートフォン（iOS 12） スマートフォン（Android 8）
ユーザ B	ノート PC（OSX）
ユーザ C	ノート PC（OSX）
ユーザ D	ノート PC（Windows 10/Debian 9）

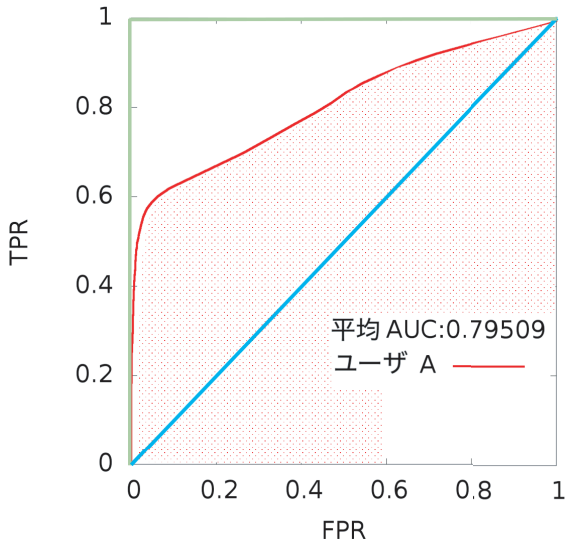

図 6 ROC 曲線の例
Fig. 6 An example of ROC curve.

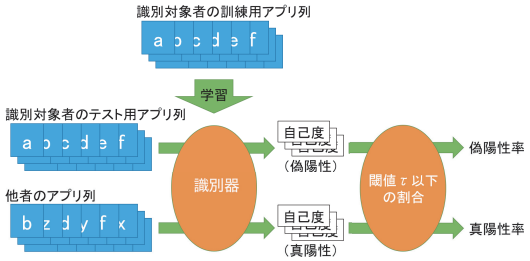

図 7 偽陽性率と真陽性率の算出の流れ

Fig. 7 Flow of calculating false positive rate and true positive rate.

己度は全体的に小さな値になることが想定される．

各通信アプリケーション列が識別対象者のものであるか否かの識別は，出力された自己度が閾値 τ より大きいか否かで判断する．自己度が閾値 τ より大きければ識別対象者自身の通信であると判定し，逆に閾値 τ より小さければ他者の通信であると判定する．よって，閾値 τ が決定すると，識別対象者自身の通信にもかかわらず他者の通信と識別した比率の偽陽性率と，他者の通信を正しく他者の通信であると識別した比率の真陽性率を得ることができる．この τ を媒介変数とすることで ROC 曲線を作成する．本研究では媒介変数として用いる自己度の閾値 τ を -0.6 から 0.6 まで 0.01 単位で変化させることにより作成した．

図 6 の赤線が具体的な ROC 曲線例である．理想的な ROC 曲線は偽陽性率が 0 のときに真陽性率が 1 になる緑線のような線である．一方で識別がまったく行えず，ランダムに結果を出力した場合，真陽性率と偽陽性率は同じように推移するため，図 6 の水色線のような対角線になる．ROC 曲線は基本的にこの水色線から離れれば離れるほど，識別性能が高いといえる．

AUC（Area Under the ROC Curve）とは ROC 曲線下の面積のことで，機械学習の識別性能を評価するためによく使われる指標の 1 つである．たとえば，図 6 の赤線の AUC は赤線下の赤点領域であり，この面積は 0.79509 である．この AUC は図の右下に表記している．理想的な ROC 曲線（緑線）の場合，その AUC は 1.0 になる．一方で，識別がランダムに行われる場合，最悪な ROC 曲線は水色線となり，この AUC は 0.5 である．

この AUC の値が 0.5 のときは識別性能がまったくなく，0.7 以上 0.8 未満の場合は識別性能があり，0.8 以上 0.9 未満の場合は優れた識別性能があり，0.9 以上 1 以下の場合は傑出した識別性能があるとされている [5]．

本研究では評価用データとして被験者以外の 4 人分のデータを使用する．その 1 人 1 人ごとに ROC 曲線を作成し，AUC を計算する．この 4 つの AUC の平均をもって識別器の性能評価に利用した．

6. 実験

提案手法の有用性を確認する実験を行った．本論文では，個人の通信のアプリケーション識別のログから，その個人の通信の振舞いを学習して作成した識別器を用いて，入力された通信アプリケーション列が本人によるものか否かを識別する手法を提案している．著者らの知る限り，個人の通信のみを抽出し，その通信が過去の自分が行った通信と比較して類似しているか否かを判定する手法がないため，相対的な評価ができない．ここでは，5.2 節で述べた機械学習の識別性能を評価する際によく用いられる AUC の値を測定することにより，提案手法の有効性を示す．

表 4 通信アプリケーション列の数

Table 4 Number of communication application sequence.

	送信元 IP ごとの分離あり	送信元 IP ごとの分離なし
識別対象者（訓練用）	41,207 件	40,367 件
識別対象者（評価用）	16,093 件	15,919 件
ユーザ A	19,014 件	18,432 件
ユーザ B	56,602 件	55,843 件
ユーザ C	9,344 件	9,054 件
ユーザ D	19,587 件	19,123 件

6.1 概要

学習に用いる通信アプリケーション列の生成方法については，4.2 節にて述べた．この実験では，通信アプリケーション列の生成において，通信ログからユーザ ID だけで抽出するか，ユーザ ID と送信元 IP アドレスの組合せで抽出するかの違い，および，通信開始時刻順に並べ替えるかどうかの違いについて実験することとした．なお，この章では，通信ログからユーザ ID だけで抽出するか，ユーザ ID と送信元 IP アドレスの組合せで抽出するかの違いのことを，発信元 IP アドレスごとの分離の有無と呼ぶ．

本実験では，学習に用いる通信アプリケーション列の生成方法において，送信元 IP アドレスごとの分離の有無，および，通信開始時刻順への並べ替えの有無の組合せにより 4 種類のデータを生成し，これら 4 種類について，識別性能の指標となる ROC 曲線と AUC を求めた．

送信元 IP アドレスごとに分離の有無により生成される通信アプリケーション列の数が異なる．生成された通信アプリケーション列の数を**表 4** に示す．

6.2 結果

生成した 4 個の識別器の ROC 曲線を**図 8** から**図 11** に示す．これらの図において，水色の線は識別能力がまったくない場合の曲線を表している．作成した識別器に対して，識別対象者の評価用の通信アプリケーション列とユーザ A の通信アプリケーション列を組み合わせたデータを入力として与えたときの識別結果を対象として，5.2 節で示した方法を用いて作成した ROC 曲線が各図中のユーザ A の曲線である．入力として与えたデータのうちユーザ A

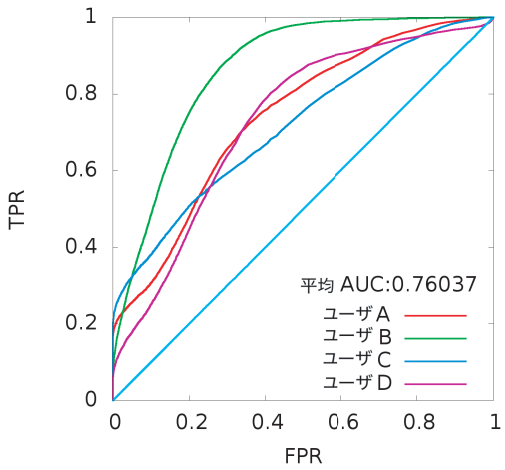

図 8 通信開始時刻順・IP 分離ありの ROC 曲線

Fig. 8 ROC Curve using data sorted by communication start time and separated by IP address.

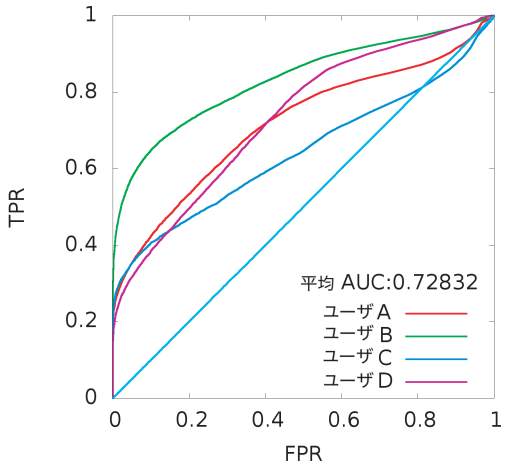

図 9 通信開始時刻順・IP 分離なしの ROC 曲線

Fig. 9 ROC Curve using data sorted by communication start time and not separated by IP address.

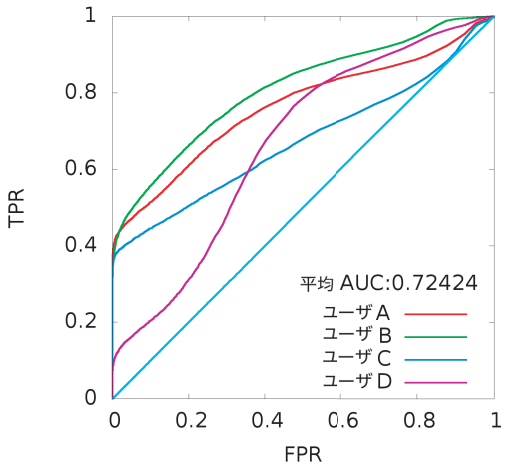

図 10 通信終了時刻順・IP 分離ありの ROC 曲線

Fig. 10 ROC Curve using data sorted by communication end time and separated by IP address.

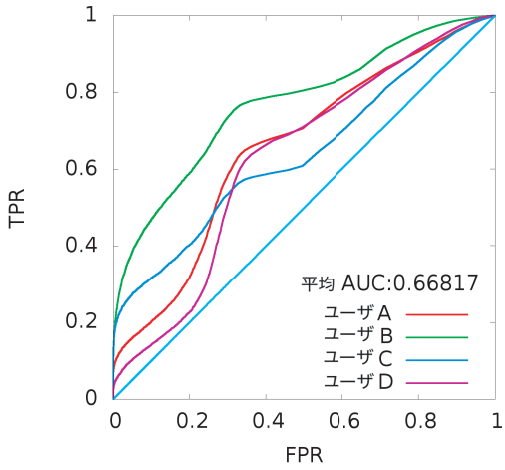

図 11 通信終了時刻順・IP 分離なしの ROC 曲線

Fig. 11 ROC Curve using data sorted by communication end time and not separated by IP address.

表 5 AUC の平均値

Table 5 The average of AUC.

	送信元 IP 分離なし	送信元 IP 分離あり
通信終了時刻順	0.66817	0.72424
通信開始時刻順	0.72832	0.76037

表 6 識別器の生成に要した時間［秒］

Table 6 The generated time of a classifier [sec].

	送信元 IP 分離なし	送信元 IP 分離あり
通信終了時刻順	1,179	1,513
通信開始時刻順	2,020	2,019

の通信アプリケーション列を，ユーザ B，ユーザ C，ユーザ D の各々の通信アプリケーション列に置き換えた場合の識別結果を対象として作成した ROC 曲線が，各図中のユーザ B，ユーザ C，ユーザ D の各々の曲線である．各図において，5.2 節に示した方法を用いて，ユーザ A，ユーザ B，ユーザ C，ユーザ D の各々の ROC 曲線を対象とした 4 つの AUC を求め，それら 4 つの AUC の平均値を求めた．その結果を**表 5** に示す．さらに，これらの識別器を作成するために必要であった学習時間を**表 6** に示す．

表 5 より，今回比較を行った通信アプリケーション列の生成方法として，通信開始時刻順に並べ替えを行い，送信元 IP アドレスごとに分離を行う場合が最も識別性能があり，そのときの AUC は 0.76037 であった（図 8）．

また 4 種類の識別器の中で最も低い AUC は 0.66817 であった．このことから今回作成した識別器であればどの手法であっても，通信アプリケーション列から，識別対象者らしさの識別能力があるものといえる．

入力となる通信アプリケーション列の作成について考察する．まず，通信開始時刻順に並べ替えるか否かに着目す

る．表 5 の並べ替え方の違いに着目すると送信元 IP アドレスごとの分離の有無に関係なく，AUC の値は並べ替えた方が良い値となっていることが分かる．

次に送信元 IP アドレスごとの分離の有無について比較する．表 5 の送信元 IP アドレスごとの分離の違いに着目すると，AUC の値は並べ替えの有無に関係なく，AUC の値は分離した方が良い値となっていることが分かる．

これらから，入力となる通信アプリケーション列の作成については，送信元 IP アドレスごとに分離して，通信開始時刻順に並べ替える方が良いことが分かる．

学習に要する計算時間，および，識別に要する計算時間について考察する．表 6 より，1 カ月分の通信アプリケーション列を学習するために必要な時間は最も長いものでも 2,944 秒であり，データの整形などの処理時間を含めても 1 時間未満であった．1 人の 1 カ月分のデータの学習が仮に 1 時間で終わるとし，30 日ごとに識別器の学習をやり直すとすると，この実験に用いた計算機 1 台を用いると 720 人に対応することが可能であることが分かった．

また，識別器に通信アプリケーション列を入力して自己度を算出する時間については，10 日分の通信アプリケーション列すべての計算に対して要した時間は 1.6 秒程度であった．識別のための計算時間は十分に短いといえる．

これらの結果より，提案手法は十分実用性があるといえる．

7. おわりに

本研究は，アプリケーション識別機能を有するファイアウォールのログを用いて，識別対象者の通信の振舞いを学習し，ある通信が自己らしい通信であるか否かを識別するための手法を提案した．

実験の結果より，ファイアウォールログから学習に使う通信アプリケーション列の生成方法として，ユーザ ID と発信元 IP アドレスの組合せで抽出し，通信開始時刻順に並べ替える方法が最も識別性能が良くなり，そのときの AUC の平均値は 0.76037 であった．これはこの識別器は識別性能を有していることを表している．

また，本研究で作成した識別器の生成時間について，最も作成に時間がかかった場合でも 1 カ月分の自己通信を学習して生成する時間は 2,944 秒であった．さらに，識別にかかる時間は 10 日分の通信アプリケーション列を対象とした場合，1.6 秒程度と非常に短い．

このことより，本研究の提案手法は十分実用性があるといえる．

今後の課題としては主に 2 点あげられる．1 つ目がより識別の精度を向上させるということである．たとえば入力データについては，今回はユーザごとに識別を行ったが物理アドレスなどを活用して，端末ごとに識別を行うことでより細かい処理を行うという方法が考えられる．ほかにも Word2vec 自体の学習の調整などが考えられる．2 つ目が有効性の検証である．今回使用したデータのユーザ数は，5 人であったため偏りがある可能性も否めない．そのため，より多くのユーザのデータと比較する実験を行う必要がある．

謝辞 本研究を行うにあたり，実験環境を提供していただいた学術情報メディアセンターと，研究を行うために必要な通信ログデータの提供に同意してくれた実験協力者に，深く感謝する．

参考文献

[1] Anderson, B., Paul, S. and McGrew, D.: Deciphering malware's use of TLS (without decryption), *Journal of Computer Virology and Hacking Techniques*, Vol.14, No.3, pp.195–211 (2018).

[2] Anderson, B. and McGrew, D.: Identifying encrypted malware traffic with contextual flow data, *Proc. 2016 ACM Workshop on Artificial Intelligence and Security*, pp.35–46, ACM (2016).

[3] Athiwaratkun, B. and Stokes, J.W.: Malware classification with LSTM and GRU language models and a character-level CNN, *2017 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp.2482–2486 (2017).

[4] Chen, Y., Zhong, K., Zhao, X., et al.: LSTM networks for mobile human activity recognition, *2016 International Conference on Artificial Intelligence: Technologies and Applications*, Atlantis Press (2016).

[5] Hosmer, D.W. and Lemeshow, S.: Area Under the ROC Curve, *Applied Logistic Regression*, John Wiley and Sons, 2nd edition, chapter 5.2.4, pp.160–164 (2000).

[6] Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, *Neural Comput.*, Vol.9, No.8, pp.1735–1780 (1997) (online), available from ⟨http://dx.doi.org/10.1162/neco.1997.9.8.1735⟩ .

[7] 神薗雅紀，遠峰隆史，井上大介ほか：プロセスの通信手続きに基づくフォレンジック手法の提案，コンピュータセキュリティシンポジウム 2014 論文集，Vol.2014, No.2, pp.167–174 (2014).

[8] Mikolov, T., Sutskever, I., Dean, J., et al.: Distributed representations of words and phrases and their compositionality, *Advances in Neural Information Processing Systems*, pp.3111–3119 (2013).

[9] 三村聡志，佐々木良一：プロセス情報と関連づけたパケットを利用した不正通信原因推定手法の提案，マルチメディア，分散協調とモバイルシンポジウム 2014 論文集，Vol.2014, pp.1973–1980 (2014).

[10] 内閣サイバーセキュリティーセンターサイバーセキュリティ戦略本部：サイバーセキュリティ研究開発戦略, 入手先 ⟨https://www.nisc.go.jp/active/kihon/pdf/kenkyu2017.pdf⟩（参照 2019-09-10）.

[11] 大倉有喜，大月勇人，田中恭之ほか：マルウェア解析のためのシステムコールトレースログと通信の対応付け手法，コンピュータセキュリティシンポジウム 2015 論文集，Vol.2015, pp.1379–1386 (2015).

[12] Pascanu, R., Stokes, J.W., Thomas, A., et al.: Malware classification with recurrent networks, *2015 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp.1916–1920 (2015).

[13] Pouyanfar, S., Chen, S. and Shyu, M.: Deep spatio-temporal representation learning for multi-class imbal-

anced data classification, *2018 IEEE International Conference on Information Reuse and Integration* (*IRI*), pp.386–393, IEEE (2018).
[14] Prasse, P., Machlica, L., Pevný, T., Havelka, J. and Scheffer, T.: Malware Detection by Analysing Encrypted Network Traffic with Neural Networks, *Machine Learning and Knowledge Discovery in Databases*, Ceci, M., Hollmén, J., Todorovski, L., Vens, C. and Džeroski, S. (Eds.), pp.73–88, Springer International Publishing (2017).
[15] Radford, B.J., Apolonio, L.M., Simpson, J.A., et al.: Network traffic anomaly detection using recurrent neural networks, arXiv preprint arXiv:1803.10769 (2018).
[16] Tobiyama, S., Yamaguchi, Y., Yagi, T., et al.: Malware Detection with Deep Neural Network Using Process Behavior, *2016 IEEE 40th Annual Computer Software and Applications Conference* (*COMPSAC*), Vol.2, pp.577–582 (2016).
[17] Wang, W., Zhu, M., Sheng, Y., et al.: Malware traffic classification using convolutional neural network for representation learning, *2017 International Conference on Information Networking* (*ICOIN*), pp.712–717 (2017).
[18] 山本 匠，河内清人，桜井鐘治ほか：不審プロセス特定手法の提案，コンピュータセキュリティシンポジウム 2013 論文集，Vol.2013, No.4, pp.634–641 (2013).
[19] Yan, J., Qi, Y., Rao, Q., et al.: Detecting Malware with an Ensemble Method Based on Deep Neural Network (2018).
[20] Yan, J., Qi, Y., Rao, Q., et al.: LSTM-Based Hierarchical Denoising Network for Android Malware Detection (2018).
[21] Zhao, Y., Yang, R., Zhang, Z., et al.: Deep residual bidir-LSTM for human activity recognition using wearable sensors, *Mathematical Problems in Engineering*, Vol.2018 (2018).

市之瀬 樹生

2016 年徳山工業高等専門学校情報電子工学科卒業．2018 年筑波大学情報学群メディア創生学類卒業．2020 年筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻博士前期課程修了．同年よりソフトバンク株式会社に勤務．

佐藤 聡 （正会員）

1996 年筑波大学大学院工学研究科単位取得退学．同年広島市立大学情報科学部助手．2001 年筑波大学講師．2013 年より筑波大学システム情報系准教授（学術情報メディアセンター勤務）．博士（工学）．主に，キャンパスネットワークの企画管理運用に関する研究に従事．

新城 靖 （正会員）

1965 年生．1988 年筑波大学第三学群卒業．1993 年筑波大学大学院工学研究科博士課程修了．同年琉球大学工学部情報工学科助手．1995 年筑波大学電子・情報工学系講師，2003 年同助教授，2004 年同大学院システム情報工学研究科助教授．2007 年同准教授．オペレーティング・システム，並行システム，仮想化，情報セキュリティの研究に従事．博士（工学）．ACM，IEEE，USENIX，日本ソフトウェア科学会各会員．

三宮 秀次 （正会員）

2002 年高知工科大学工学部情報システム工学科卒業．2006 年高知工科大学助手．2007 年同大学大学院博士課程単位取得後退学．2010 年筑波大学助教，現在に至る．博士（工学）．低消費電力化データ駆動プロセッサの研究に従事．電子情報通信学会，IEEE 各会員．

星野 厚

2006 年筑波大学大学院理工学研究科修士課程修了．株式会社チノウ代表取締役．現在，筑波大学システム情報工学研究科コンピュータサイエンス専攻に在学中．人や社会と人工知能の関係に興味がある．