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くない通信の識別手法

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

概要：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ふるまいを学習し，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に有効な手法を提案する．入力としてファイアウォールが解析した通信アプリケーション列が与えられた際に，識別対象者らしくなさを表す数値を出力する識別機を作成した．この識別機に 8 種類の入力形式のデータを用いて識別性能のテストを行い，どの入力データ形式の際に高い識別性能を出せるかを調査した．結論として，送信元 IP アドレスごとに分離した，DNS を削除した通信アプリケーション列，Embedding は word2vec を用いることで，高い識別性能が実現できることが分かった．

Identification method of network traffic unlikely belonging to target users by machine learning using application-identifying firewall log

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

Abstract: The purpose of this study is to clarify an effective method that can identify whether network traffic belongs unlikery to target users or clients than the others. We realized a classifier that learns target user's network traffic behavior by using the log of firewall system with application identification capability. In the learning, we fed communication applications sequence extracted from the log. The classifier outputs numeric value representing the degree of the uncertainty that indicates that the traffic belongs to or generated by the target user. To evalute this classifier, we investigated its classification acuracy by using input data formatted by eight different types and found an input data format type that leads to the hiest accuracy. According to its results, it is revealed that we can realized highly accurate classifier by separating and grouping input data for each source IP address after deleting the DNS records of them, while using word2vec as Embedding.

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 [9].

本研究では，この脅威の一部は何らかの普段と異なる通信を行うと仮定する．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ユーザが行った通信などである．この様に，普段と異なる，自己らしくない通信が行われていることを検出することができれば，その通信を行ったコンピュータに何らかの脅威が，存在する

1 筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba
2 筑波大学システム情報系情報工学域
Department of Information Engineering, Faculty of Engineering, Information and Systems, University of Tsukuba
3 筑波大学学術情報メディアセンター
Academic Computing and Communications Center, University of Tsukuba
4 株式会社 チノウ
Chinoh, Ink

可能性を警告することができる．

本研究は，識別対象者の通信のふるまいを学習し，ある通信が自己らしい通信であるか否かを識別する．そのための手法として，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ることを提案する．

従来のふるまい検知等では監視用のソフトウェアを，各コンピュータにインストールすることが必須であった．この方法と比較して，提案手法の利点は，インストールを強制できない環境や，監視対象のコンピュータが多い場合などでも，導入することが比較的容易であることである．また近年，爆発的に増加している一方でセキュリティ対策が困難なため，大規模な感染を起こした IoT 機器 [15] においても適用できる．

本研究では，ある通信が識別対象者の通信らしいか否かを，識別できる識別機を作成する．本論文で作成する識別機は，ファイアウォールログが入力された際に，識別対象者らしさを出力する．識別機の精度は，ある識別対象者のログで学習させた識別機に，識別対象者自身のログを入力した場合と，他のユーザのログを入力した場合とで，出力される識別対象者らしさに，差があるかどうかで評価する．また，様々なログの前処理方法を試し，有効な方法を示す．

2. 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ふるまいを学習し，ある通信が自己らしい通信であるか否かを識別する．そのために，入力にファイアウォールのログを前処理して与えた際に，識別対象らしさを出力する識別機を作成する．また，本研究では様々な手法でファイアウォールログの前処理を行い，どの手法が最も高い精度で識別できるかについて調査する．

2.1 想定する環境

2.1.1 ファイアウォール

本提案手法が想定する環境では，アプリケーション識別機能がついたファイアウォールを用いる．この機能は通過したパケットをファイアウォールが解析し，その通信を行ったアプリケーションの名前を出力するものである．アプリケーションの名前としては，単に Web だけでなく，Google や Twitter 等のサービス名も出力できる．この機能を用いることにより，従来のパケットを解析する手法と比較して，複雑さを減らすことができる．従来の手法では宛先 IPv4 アドレスや宛先ポート番号などから，マルウェア検出を行ってきた [14]．しかし宛先 IPv4 アドレスと宛先ポート番号の組み合わせは，$2^{32} \times 2^{16} \approx 281$ 兆種類である．一方でアプリケーション識別機能を有するファイアウォールが識別するアプリケーションの名前は数千種類程度である．よって宛先 IPv4 アドレスや宛先ポート番号を使用するのと比べて，アプリケーション名を用いることで大きく種類数を削減でき，少ない計算資源を用いて短い解析時間で解析可能となる．

2.1.2 ネットワーク構成

本提案手法で想定するネットワーク環境の概要を図 1 に示す．

2.1.2.1 ユーザ・端末識別

本提案手法ではユーザごとの通信アプリケーション列を用いて，その通信アプリケーション列が自己らしいか否かを学習する．そのため，ファイアウォールのログを何らかの方法で，ユーザごとに分別できる必要がある．

DHCP を用いずに固定 IP アドレスのみを用いるネットワークであれば，IP アドレスをそのまま用いてユーザを識別することが可能である．DHCP による IP アドレスの割り当てを行っていた場合は，DHCP サーバのログから，割り当てた IP アドレスとその対象の端末の MAC アドレスを取得することで，端末の識別が可能である．また，より小規模なネットワークであれば MAC アドレスに対して，払いだす IP アドレスを固定する等により，IP アドレスによるユーザの識別も可能である．

2.1.2.2 ファイアウォール設定

ファイアウォールの設定としては外向き通信は許可，内向き通信は拒否とする．この設定は NAPT（Network Address Port Translation）の機能とほぼ同じであり，企業のネットワークなどでは一般的な設定と言える．

2.2 識別機の作成

本提案手法では識別機をディープラーニングによって作成する．機械学習の入力としては，ファイアウォールログから，アプリケーション名を抽出したデータを用いる．これを以降アプリケーション列と呼ぶ．ウインドウサイズとしては，現在 30 としている．

2.2.1 DNS

ファイアウォールによって識別されるアプリケーションには，DNS アクセスが含まれる．しかし，2 つの理由からユーザの特徴を学習することにおいてノイズになる可能性があると考えられる．1 つ目が DNS アクセスによるログの数が他のアプリケーションに比べて多いことである．これ

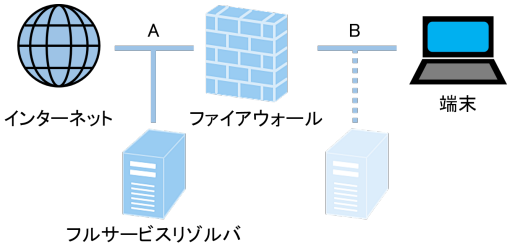


図 1 想定する大まかなネットワークの構成

は，あるユーザ*1の 1 か月分のアプリケーション列の出現比率を予備調査した結果から分かる．このアプリケーション比率を図 2 に示す．2 つ目がユーザの意思に関係なく自動で行われる通信であることである．これらより，DNS を除外するか否かで検出性能に差が出るかを調査する．

また，本提案手法で想定しているネットワークは図 1 のような構成であり，フルサービスリゾルバが図 1 の A の位置に設置されているため，DNS の通信についても記録することができる．一方フルサービスリゾルバを，B の位置に設置する場合，DNS に関連した通信をファイアウォールで検出できないことも，DNS の有無の差を調査する理由の一つである．

2.2.2 Embedding

本研究では，Embedding の違いによる性能差を調べるため，one-hot 表現と word2vec の二つを比較する．

2.2.2.1 one-hot 表現

one-hot 表現とは各次元に一つの意味を割り当てることで表現する方法である．例えばアプリケーションの種類が 3 つあった場合，それぞれ [1,0,0]，[0,1,0]，[0,0,1] のように 1 つの次元だけ 1 にし，それ以外の次元を 0 にしたベクトルで表現する．

本来はファイアウォールが識別可能なアプリケーションすべてに，対応するベクトルを用意する必要がある．しかし，識別対象者によっては全く利用しないアプリケーションもある．そこで本提案手法では，表現するアプリケーション数を減らし，次元の削減を行う．具体的には識別対象者の教師データにおいて，(現れるアプリケーションの種類数+1) 次元のベクトルに変換する．追加した 1 次元は，識別対象者の教師データには現れなかったアプリケーションを表現する．

2.2.2.2 word2vec

word2vec は，自然言語処理の分野で成果を出している手法 [7] である．単語の前後のつながりを学習することで単語のベクトル化を行う．

word2vec では似た意味の単語間の距離は小さく，逆に大きく意味が異なる単語間の距離は大きくなる．このことから例えば twitter と facebook は，同じ SNS のアプリケーションであることから，似たベクトルが生成されることが想定される．これはすべての単語間のベクトルの距離が同じ，one-hot 表現とは大きく異なる．また，自然言語の世界では，例えば「king - man + woman = queen」のように，単語の意味同士の足し引きが行えるという特徴がある．

2.2.3 ニューラルネットワーク

本研究で用いるニューラルネットワークは RNN (Reccurent Neural Network) である．RNN は通常のニューラルネットワークと比較して，隠れ層にループのような構造を持っている．そのため，RNN は時系列方向に複数の入力があった際に，1 つ前の隠れ層の状態を加味した計算が可能で，時系列データの処理を行うことができるという特徴がある．また，今回は RNN の中でも LSTM (Long Short-Term Memory) と呼ばれる，隠れ層の代わりに LSTM ユニットという特殊なセルを活用する方法を用いる．

2.3 評価方法

本提案手法では最終的にある通信アプリケーション列に対して，その列がどれほど自己らしくないかを表す値を出力する．本論文はこの値を**他者度**と呼称する．それをもとに ROC (Receiver Operatorating Characteristic) 曲線を描き，AUC (Area Under the Curve) を算出する．この AUC の値をもとに識別機の比較を行う．

2.3.1 ROC 曲線

ROC 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 3 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てしまった比率，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．

閾値 τ が与えられたときに，識別対象者の通信を入力と

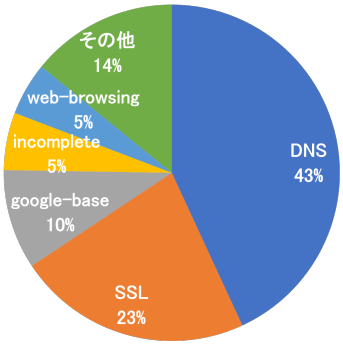


図 2 通信アプリケーション比率

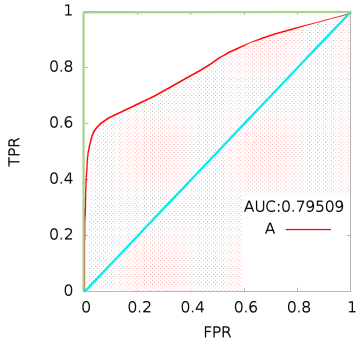


図 3 ROC 曲線の例

*1 このユーザのデータは 4 章において識別対象者のデータとして利用した．

して出力された他者度のうち τ 以上の，通信を自己らしくない通信と識別した割合である偽陽性率と，識別対象者でない通信を入力として出力された他者度のうち τ 以上の，通信を自己らしくない通信と識別した割合である真陽性率が得られる．この τ を媒介変数とすることで曲線を描く．

図 3 の赤線が具体的な ROC 曲線例であり，理想的な ROC 曲線は偽陽性率が 0 の時に真陽性率が 1 になる，緑線のような線である．ROC 曲線ではその識別性能を AUC と呼ばれる曲線下の面積で表現する．

2.3.2 AUC

AUC は機械学習の識別性能を評価するために，使われることが多い指標の一つである．例えば，図 3 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．これは図の右下に表記している．これが理想的な ROC 曲線（緑線）の場合，その AUC は 1.0 になる．もし，識別がランダムに行われる場合，真陽性率と偽陽性率がほぼ同じ値で推移するため，水色の斜線上の線となり，この AUC は 0.5 である．つまり AUC が 0.5 以上，偽陽性率が真陽性率を上回っている場合，何らかの識別が行えていると言うことができる．本研究ではこの AUC をもとに識別機の性能を評価する．

3. 実装

本実装は本学の認証付き無線 LAN のネットワークを想定して行った．認証付き無線 LAN のネットワーク構成は図 1 と同様である．このネットワークを利用するためには認証を行う必要がある．また，外向き通信は許可されているが内向き通信は拒否されている．

3.1 使用するファイアウォールとログの項目

本研究で利用した具体的なファイアウォールは，本学で運用されている，Palo Alto Networks 社製のファイアウォール，version 7.1 である．本論文で利用したログは，ログデータの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，各自の通信ログを開示してもらったものを，著者らに提供していただいたものを利用した．このログの中で使用した項目は，通信開始時刻，通信したアプリケーション，送信元 IP アドレスの 3 つである．それぞれの項目について要件を示す．

3.1.1 通信開始時刻

本学のファイアウォールは通信終了時に，アプリケーション名をログに出力する．一方，提案手法では開始時刻の順に処理したい．そこで，このログを開始時刻順に並び替える．開始時刻はこのログの順番を並び替えるためだけに用いており，他者度を算出するためには利用していない．

3.1.2 通信したアプリケーション

本学が導入しているファイアウォールでは DNS や 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 2962 種から決定される．それに加え，本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として定義している．

3.1.3 送信元 IP アドレス

ユーザごとのログデータをそのまま用いると，1 人のユーザが複数の端末を同一時間帯に使用していた場合，異なる複数の端末の通信が，一つの連続した通信アプリケーション列として，処理されてしまう恐れがある．しかし，このようなデータは通信したタイミングによって，通信アプリケーション列の組み合わせが変わることから，解析することは困難である．そこで各ユーザの通信において，ファイアウォールログの送信元 IP アドレスごとに分離することを考えた．

ファイアウォールログには送信元 IP アドレスが記録されている．この送信元 IP アドレスは，本学のネットワークにログインした際に端末ごとに割り当てられるもので，同時刻に同じものは存在しない．そのため送信元 IP アドレスごとに分離することによって，複数の端末の通信アプリケーション列が，1 つの連続した通信アプリケーション列として処理されることを防ぐことができる．この項目は初めから IP アドレスをもとにユーザを識別している場合や，MAC アドレスを基準にログをとっているような環境であれば必要ない．

また，認証付き無線 LAN システムでは，DHCP による IP アドレスの割り当てを行っている．DHCP のリリース時間を超過した際に，すでに利用をやめていた場合には，IP アドレスが回収される．そのため，送信元 IP アドレスに着目して通信アプリケーション列を分離すると，連続した通信アプリケーション列の間に長い時間が経過し，連続性がない場合に，それらを分離することが期待できる．

3.2 ファイアウォールログの前処理

本研究におけるファイアウォールログの前処理の概略を図 4 に示す．本提案手法の前処理では主に以下の 2 点を考

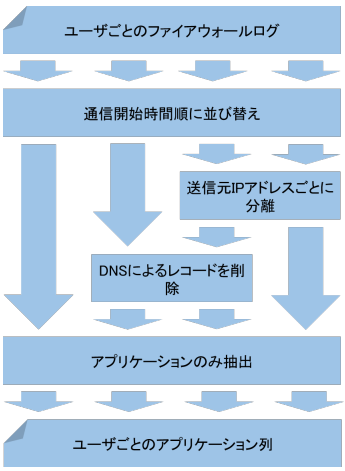


図 4 データ加工の流れ

慮した．一つ目がアプリケーションの DNS を含めるか否か，二つ目が送信元 IP アドレスごとに分離するか否かである．最終的にこれらの入力データの中で，最も高い識別精度が出るものを調査する．

本研究で使用するファイアウォールログは，提供時までに通信開始時刻, 通信したアプリケーション, 送信元 IP アドレスの 3 カラムが抽出されている．最初にログに記録されている通信開始時刻を基準に並び替える．並び替える際に同時刻のログデータについては，あらかじめ記録されている順序，言い換えれば通信終了時刻順になるようにした．次に送信元 IP アドレスごとに分離する場合は，送信元 IP アドレスごとに分離する．次に DNS を含めない場合は，アプリケーションが DNS と識別されているログを削除する．最後にアプリケーションのカラムのみを抽出する．ここから得られるデータは改行区切りのアプリケーション列のみである．

3.3 Embedding

3.3.1 one-hot 表現

本研究で使用する訓練データのアプリケーションの種類数は 68 種類であった．このことから今回は (68+1) 次元のベクトルで表現する．

3.3.2 word2vec

word2vec の手法で単語をベクトル化する際には，ニューラルネットワークを用いた機械学習を行う必要がある．word2vec を作成する上で用いた学習データは，識別対象者の訓練データであり，単語数が少ないものについてはまとめて，unknown という単語として学習を行った．また，他ユーザのデータセット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．本研究では 32 次元のベクトル化を行っている．また，ターゲットの単語に対して前後 1 単語を skip-gram によって学習させている．

3.4 機械学習

本研究ではある通信アプリケーション列が与えられたとき，その列が自己による通信であるか否かを学習し，他者度を出力する識別機を作る必要がある．そのため自己データとして識別対象者のアプリケーション列を用意した．また，他者データとしては識別対象者の教師データで現れたアプリケーションから，ランダムに選択して生成した，アプリケーション列を用意した．これらを教師データとして学習を行う．この際，自己の通信データ 1 割，他者の通信データ 9 割の比率で生成した．この自己と他者の通信データの比率は，予備実験において 1:9 の方が 1:1 より良い結果を，得られることが分かっているためである．しかし，1:9 が最も良い結果を得られる比率であるかどうかは，検証が不足しているため不明であり，今後の課題である．1 度に入力する通信アプリケーション列であるウインドウサイズは 30 とし，スライディングウインドウ方式で取得した．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] としてミニバッチ学習を行った．この際のバッチサイズは 200 とした．LSTM ユニットは，TensorFlow の BasicLSTMCell である．また，使用言語は Python である．機械学習は CPU に Core i5-6400, メインメモリに 8GB，GPU に GeForce GTX1050Ti を搭載した，一般的なゲーミング PC 程度のスペックのマシンを用いた．

3.5 出力

識別機は自己の通信を [1,0]，他者の通信を [0,1] として学習し，2 次元ベクトルの出力をする．本研究では出力 [a,b] が得られたとき，他者度を次のように定義する．

$$\begin{cases} 0 & (a = b = 0) \\ \frac{b}{a+b} - 0.5 & (otherwise) \end{cases} \quad (1)$$

この式を用いて通信アプリケーション列ごとに他者度を算出する．他者度は値が正の時に他者らしい通信，負の時は識別対象者らしい通信であることを表す．他者度の絶対値が大きければ大きいほど確信度が高くなる．この他者度がある閾値を上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もできる．またこの他者度の分布をもとに識別機の評価を行う．本研究での識別機の評価の流れを図 5 に示す．

4. 実験

4.1 概要

ファイアウォールログデータの提供の同意を得た 5 人分の，ログデータを用いて比較実験を行った．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．本実験で用いたログの件数を表 1 に示す．

DNS の有無，送信元 IP アドレスごとの分離の有無，Embedding の方式（one-hot か word2vec）を比較するため，本実験では入力データの形式 4 種と，Embedding 形

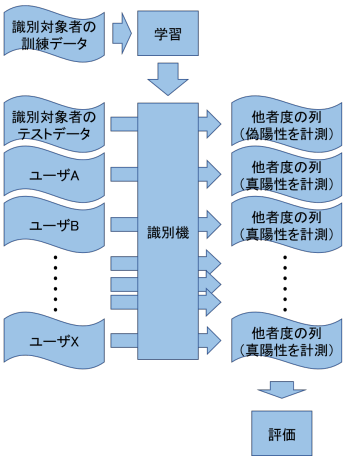


図 5 評価の大まかな流れ

表 1　本研究で用いたログの件数

	DNS あり	DNS なし
訓練データ	75,608 件	41,237 件
テストデータ	25,203 件	16,123 件
ユーザ A	30,247 件	19,044 件
ユーザ B	66,054 件	56,632 件
ユーザ C	15,017 件	9,374 件
ユーザ D	35,252 件	19,617 件

表 2　識別機の AUC

(a) one-hot 表現を用いた識別機の AUC

	DNS あり	DNS なし
送信元 IP アドレス分離なし	0.66658	0.68353
送信元 IP アドレス分離あり	0.63112	0.72898

(b) word2vec を用いた識別機の AUC

	DNS あり	DNS なし
送信元 IP アドレス分離なし	0.64015	0.72832
送信元 IP アドレス分離あり	0.65132	0.76037

式 2 種の 8 パターンを試行する．これら 8 パターンについて，識別性能の指標となる ROC 曲線と AUC を求める．

ROC 曲線を描く際には，媒介変数として用いる他者度の閾値 τ を-0.6 から 0.6 まで 0.01 単位で変化させた．また，今回は識別対象者以外のユーザのデータとして，4 人分のデータを用いている．各識別機の AUC はこの 4 人の ROC 線の AUC の平均をとるものとした．

4.2　結果

生成した 8 つの識別機の ROC 曲線を図 6 に，AUC を表 2 に示す．表 2 より，今回比較を行った前処理の中では送信元 IP アドレス分離ありで DNS なし，word2vec を用いたものが最も優秀であることがわかった（図 6(h)）．図 6(h) に着目すると，低偽陽性率の領域で高い真陽性率を得ることができている．具体的には，偽陽性率 0%に着目すると検出しにくいユーザ D のログでも 15%，検出しやすいユーザ B の通信では 40%他者であると識別することができた．

また，すべての識別機の ROC 曲線について，AUC が 0.5 以上となっており，通信アプリケーション列のみから，その列が識別対象者らしいか否かの識別を行うことが，ある程度は可能であるということが分かる．

次に DNS あり（図 6(a),6(c),6(e),6(g)）と DNS なし（図 6(b),6(d),6(f),6(h)）の場合について，比較する．表 2 で DNS ありの左列と DNS なしの右列を比べるとすべての組み合わせにおいて，DNS なしの方が DNS ありよりも AUC が大きく，有効であることが分かる．次に送信元 IP アドレスをもとにした分離あり（図 6(c),6(d),6(g),6(h)）となし（図 6(a),6(b),6(e),6(f)）について比較する．表 2 で分離ありの上段と分離なしの下段を比べると，DNS あり，one-hot 表現の組み合わせを除いた，3 つの組み合わせにおいて送信元 IP アドレス分離なしに比べて送信元 IP アドレス分離ありのモデルの方が高い AUC を示している．最後に one-hot 表現（図 6(a),6(b),6(c),6(d)）と word2vec（図 6(e),6(f),6(g),6(h)）について比較する．表 2(a) と表 2(b) を比べると，送信元 IP アドレス分離なし，DNS ありの組み合わせを除いた，3 つの組み合わせにおいて one-hot 表現よりも word2vec のモデルの方が高い AUC を示している．

以上のことをまとめるたものを以下に示す．

- DNS ありより DNS なしの方が優秀
- one-hot 表現より word2vec の方が優秀
- 送信元 IP アドレスごとに分離しないよりした方が優秀

また，これら識別機を作成するために，1 か月分のデータを学習するのに要した時間は，最も長いものでも 2612 秒であり，データの整形等の処理時間を含めても 1 時間未満であった．一方で 10 日分のアプリケーション列を入力した際に，それぞれの他者度を算出するのに要した時間は 1.6 秒程度であり，30 個のアプリケーション列 1 つに対する判定時間は十分に短い時間である．この結果より，本提案手法は計算時間的に十分実用性があるといえる．

5.　関連研究

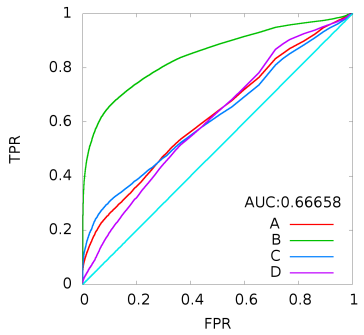
5.1　通信情報を用いたマルウェア検知

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][3][12][14][17]．これらのマルウェアの検知器は，汎用的に使える．本研究では，これらの検知器では検出できない，端末の乗っ取りなど他ユーザによる通信を検出できる．また，これらの検出器では学習を行う際にマルウェアの通信データが必要になるが本研究では必要ない．

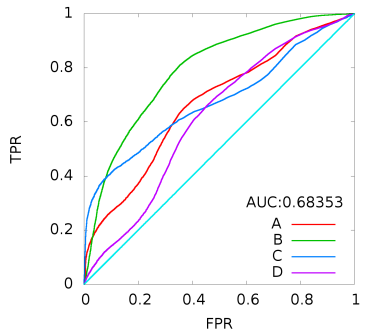
暗号化されたネットワークトラフィックから TLS ハンドシェイク，DNS コンテキスト，HTTP コンテキストなどのメタデータを用いたり，HTTPS に特化したり，バイナリ分布を用いることで，マルウェアを検知できることが示されている [1][2][12][17]．これらはパケット，または TLS メッセージ単位の詳細な解析が必要である．そのため本研究で用いるアプリケーション名のみの場合と比較して，ログデータの保管領域や解析資源を多く必要とする．

また，通信情報を用いたマルウェア検知については，通信自体が時系列データであるということもあり，RNN，LSTM[5] を用いることで，検出できることが示されている [3][12][14]．本研究でも LSTM を利用することで成果が良い識別機が作成できる．

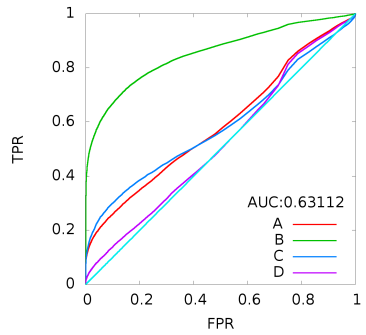
マルウェアの動作は実際に端末の通信に影響を及ぼすことを確認するため，それを発見するための手法が研究 [8][10] されており，プロセスと通信の傾向に関連があることも示されている [6]．このことからその端末の内部の情報が普段通りかそうでないかの 2 値分類のみである本研究は難易度が低く，より高い精度で分類できると考えられる．



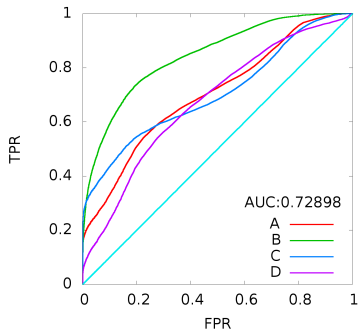
(a) IP 分離なし・DNS あり・one-hot 表現の ROC 曲線



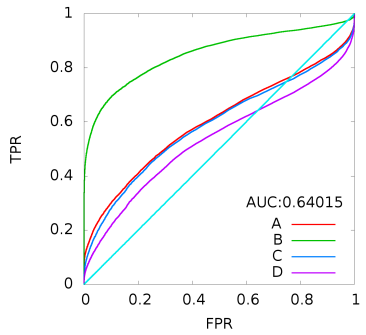
(b) IP 分離なし・DNS なし・one-hot 表現の ROC 曲線



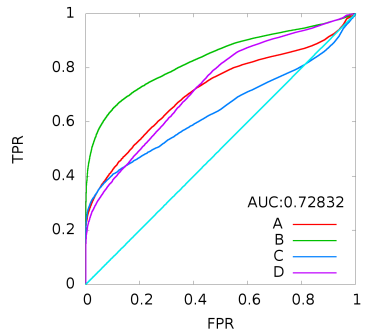
(c) IP 分離あり・DNS あり・one-hot 表現の ROC 曲線



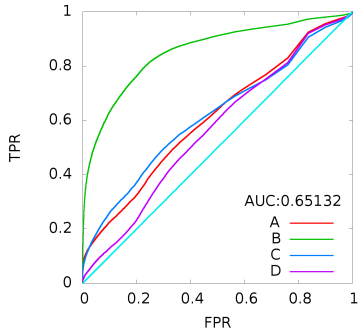
(d) IP 分離あり・DNS なし・one-hot 表現の ROC 曲線



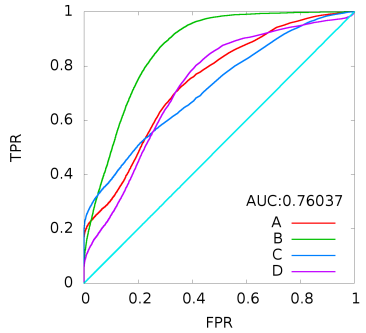
(e) IP 分離なし・DNS あり・word2vec の ROC 曲線



(f) IP 分離なし・DNS なし・word2vec の ROC 曲線



(g) IP 分離あり・DNS あり・word2vec の ROC 曲線



(h) IP 分離あり・DNS なし・word2vec の ROC 曲線

図 6 出力された ROC 曲線

5.2 LSTM を用いたマルウェア本体の検知

感染を未然に防ぐためにパターンマッチングを機械学習等を持ちいて，柔軟に行う方法も研究されている．

例えば 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等を使用することで，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 [11][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ファイル本体が必要であり，本研究と異なり外部からマルウェアに感染しているかを判断することはできない．

5.3 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ーの情報から，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，ウェアラブルセンサーから得た情報をもとに双方向 LSTM を用いて活動認識を行っている．ウインドウサイズは主に 500ms から 5000ms である．

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．40 フレームのウインドウサイズを用いて学習を行っている．

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た．ウインドウサイズは 50 である．

これらの研究を参考にして，本研究ではウインドウサイズが 1 秒以下のものが 1%未満となるように調整した．

6. 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ふるまいを学習し，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．

その結果，ファイアウォールログから取得した時系列順に並べたアプリケーション列の情報を，自己の通信の識別に用いることの有用性を提示した．またファイアウォールログの前処理において，DNS を含めるか否か，Embedding 方式は word2vec と one-hot 表現のどちらが良いか，送信元 IP アドレスごとに分離すべきか否か，調査した．結果，最も優秀な識別性能を示した前処理は，アプリケーション列から DNS を除き，送信元 IP アドレスごとに分離し，word2vec を用いた方法で，特に低偽陽性率の領域で高い真陽性率を得られることが分かった．

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．例えば入力データについては，送信元 IP アドレスごとに分離した方が良かったという結果から，MAC アドレスごとに分離するという方法が考えられる．他にも word2vec の学習の調整，一度に入力するログの件数の調整などである．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ことで，確かに有用であると示す必要があると言える．

謝辞 本研究を行うにあたり，必要な通信ログデータを提供していただいた学術情報メディアセンターと，提供に同意していただいたソフトウェア研究室の皆様に，この場を借りて深く感謝いたします．

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, *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security*, ACM, pp. 35–46 (2016).

[3] Athiwaratkun, B. and Stokes, J. W.: Malware classification with LSTM and GRU language models and a character-level CNN, *2017 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2482–2486 (2017). ID: 1.

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

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

[6] 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).

[7] 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).

[8] 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).

[9] 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，`https://www.nisc.go.jp/active/kihon/pdf/kenkyu2017.pdf`. Accessed:2019-09-10.

[10] 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).

[11] Pascanu, R., Stokes, J. W., Thomas, A. et al.: Malware classification with recurrent networks, *2015 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1916–1920 (2015). ID: 1.

[12] Paul, P., Machlica, L., Tomavs, P. et al.: *Malware Detection by Analysing Encrypted Network Traffic with Neural Networks*, pp. 73–88 (online), available from ⟨https://doi.org/10.1007/978-3-319-71246-8_5⟩, Springer Int. Publishing (2017).

[13] Pouyanfar, S., Chen, S. and Shyu, M.: Deep spatio-temporal representation learning for multi-class imbalanced data classification, *2018 IEEE Int. Conf. on Information Reuse and Integration (IRI)*, IEEE, pp. 386–393 (2018).

[14] 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).

[15] 総務省：平成 30 年版情報白書，`http://www.soumu.go.jp/johotsusintokei/\\whitepaper/ja/h30/pdf/n6500000.pdf`. Accessed:2019-07-04.

[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 Conf. (COMPSAC)*, Vol. 2, pp. 577–582 (2016). ID: 1.

[17] Wang, W., Zhu, M., Sheng, Y. et al.: Malware traffic classification using convolutional neural network for representation learning, *2017 Int. Conf. on Information Networking (ICOIN)*, pp. 712–717 (2017). ID: 1.

[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).