Computer Security Symposium 2020
26 - 29 October 2020

深層学習を用いた暗号化 SMB 通信からのリモートコマンド推定

江田　智尊[1,a)]　小久保　博崇[1]　海野　由紀[1]　森川　郁也[1]　村上　雅彦[1]

概要：ネットワークセキュリティにおいて，インシデント発生後の迅速な全容把握は被害の最小化に重要な役割を果たす．全容把握の一材料として，攻撃者やマルウェアなどが実行した悪意のあるリモートコマンドを証跡化することが求められる．我々の従来研究では，リモートコマンド実行時の通信パケットから端末で実行されたコマンドを推定し証跡化する技術を開発した．しかし SMB プロトコル version 3 の普及により通信パケットが暗号化されるようになり，コマンドを証跡化できない問題が生じた．

本論文では，リモートコマンド実行時の暗号化された SMB を含む通信パケットから，端末で実行されたコマンドを推定する深層学習技術を提案する．本技術は SMB メッセージサイズの系列を入力，コマンド名を出力とする畳み込みニューラルネットワークを基にする．技術の工夫により，従来より効率的な小標本での学習と，コマンド推定根拠の可視化を実現する．実験の結果，97.8 % の精度で暗号化通信からのリモートコマンド推定を実現した．

キーワード：暗号化通信解析，機械学習，深層学習，フィンガープリンティング，フォレンジック

Remote Command Fingerprinting on Encrypted SMB Traffics with Deep Learning

Satoru Koda[1,a)]　Hirotaka Kokubo[1]　Yuki Unno[1]　Ikuya Morikawa[1]　Masahiko Murakami[1]

Abstract: It plays an important role to instantly reveal the scenario of cybersecurity attacks after incidents for mitigating the damage caused by the attacks. Our previously developed forensic tool realizes the tracing of malicious remote commands executed by attackers or malwares by analyzing network packets. Due to the spread of the SMB protocol version3 which has the function of packet encryption, however, the command tracing becomes impossible.

This study proposes a remote command fingerprinting approach based on deep learning which enables us to infer executed remote commands even from encrypted packets. Its building block is the convolution neural network, whose inputs and outputs are sequences of SMB message size and command identifiers, respectively. Additionally, our approach makes it possible to efficiently train the network model on small samples and visualize the reason of the inference. Our experiments show that our approach can realize the remote command fingerprinting with the accuracy of 97.8%.

Keywords: Deep Learning, Digital Forensics, Encrypted Traffic Analysis, Machine Learning, Traffic Fingerprinting

1. はじめに

研究背景：標的型攻撃が巧妙化している．攻撃者は組織内ネットワークの端末を RAT（Remote Access Trojan/Remote Administration Tool）と呼ばれるマルウェアに感染させ，RAT を介して端末をリモート操作し感染拡大や機密情報の収集を行う．攻撃者は巧妙に標的ネットワークに侵入し，OS 標準のコマンドを使うなどして通常の業

[1] 株式会社富士通研究所 セキュリティ研究所
Security Research Laboratory, Fujitsu Laboratories Ltd.
[a)] koda.satoru@fujitsu.com

務に紛れて目立たないような攻撃活動を行う．このような巧妙な攻撃を完全に防ぐことは難しく，セキュリティインシデントを未然防止することは不可能な状況にある．このため近年では，マルウェア感染を防ぐ防御技術に加え，感染後の事後対応のレスポンス向上を意図した技術が強く求められている．このような技術を用いることで，感染被害が拡大する前に攻撃全容を把握し適切な対処を実施することが可能になり，甚大な損害が発生することを防止することができる．

先行研究：全容の把握には，攻撃者がネットワーク機器や端末に残した痕跡を解析するデジタルフォレンジックと呼ばれる方法が採られている．我々の先行研究では，攻撃者が行ったリモート操作と悪用されたアカウントに着目して攻撃証跡を収集し，収集した証跡から攻撃の進行状況を分析して被害範囲を高速に特定するネットワークフォレンジック技術を開発した [1]．本技術の一要素であるリモート操作証跡化技術は，攻撃者やマルウェアによるリモートコマンド実行時に流れるパケット，特にプロセス間通信プロトコルである Server Message Block（SMB）を解析し，それらが実行したコマンドを特定する．本技術により，膨大な通信データをコマンドレベルに要約・記録して証跡ログサイズを縮小しつつ，フォレンジックに必要なリモート操作情報を保存することが可能になった．

課題：しかしながら，SMB プロトコル version 3 で SMB メッセージの暗号化機能が追加されたことにより，従来技術によるコマンド特定が不可能となった．SMB version 3 は，Windows 8, Windows Server 2012 以降の OS に標準搭載されており，容易な設定で暗号化機能をオンにすることができる．暗号化によりリモートコマンドを特定できなければ，インシデント発生後の全容把握・処置が遅れ，組織に甚大な被害をもたらすこととなる．

提案技術：そこで本論文では，リモート操作コマンド実行時の暗号化された SMB メッセージから，実行されたコマンドを推定する深層学習技術を提案する．本技術は畳み込みニューラルネットワーク (CNN: Convolutional Neural Network) を用いたコマンド判別問題として定式化される．入力にはコマンド実行時のパケットキャプチャ（pcap）に含まれる SMB メッセージサイズ系列を受け取り，出力としてコマンドの識別子を出力する．また 2 つの特徴を有する．1 つ目に triplet ネットワーク [2] を用いて，訓練サンプルが少ない状況においても精度の高いコマンド推定を実現する．2 つ目に Grad-CAM [3] を用いて，推定根拠の可視化を実現する．本特徴により，高精度で説明性のあるリモートコマンド推定を実現する．数値実験では 36 通りのリモートコマンドを対象にコマンド推定実験を行った．結果，暗号化通信からのコマンド推定を 97.8%の精度で実現した．

貢献：本論文の貢献は以下のとおりである：

- 本研究は知る限り，暗号化通信からのリモートコマンド推定の初の試みとなる．この実現により，暗号化通信を用いた標的型攻撃に対する高速なデジタルフォレンジックを可能にする．
- 暗号化通信解析でも利用可能な推定根拠を可視化する解析技術を提案する．これにより，推定モデルが何を根拠に推定を行ったかを検証することが可能になる．

論文構成：本論文の構成は以下の通りである．2 章で関連研究，3 章で関連技術をレビューする．4 章で提案手法を説明し，5 章に数値実験の結果を記述する．6 章で研究倫理，7 章で結論を述べる．

2. 関連研究

暗号化通信解析技術について関連研究を述べる．暗号化通信を解析しフォレンジックに限らずあらゆる用途に解析結果を利用する事例は多数存在する．ここではその中でも提案技術に関連の深い暗号化通信のパッシブフィンガープリンティング (FP: FingerPrinting) 技術について述べる．

2.1 暗号化通信フィンガープリンティング

暗号化通信のパッシブ FP は，受動的に観測できる暗号化通信を観測し得られる情報から，何らかの対象物を一意に特定することを目的とする．例えば IoT デバイス FP では，デバイス–サーバ間の暗号化通信のパターンから，通信を行ったデバイスを特定する [4]．他にも暗号化通信から，アクセスしたウェブサイト（[5], [6], [7], [8], [9], [10], [11]），モバイルアプリケーション（[12], [13], [14]），通信種別 ([15], [16]) などを特定する事例も存在する．これらは，暗号化通信に含まれる対象物固有の通信パターン（フィンガープリント）を見出すことで対象物の特定を実現する．

2.2 深層学習を用いた暗号化通信フィンガープリンティング

近年は暗号化通信 FP に深層学習技術が用いた事例が多数存在する．深層学習を用いた FP は一般に判別問題に帰着される．即ち，暗号化通信（入力）と対象物の識別子（ラベル）のペアを用いてこれらの関係性を学習し，暗号化通信から対象物を判別するモデルを構築する．学習によって対象物を特定するフィンガープリントを見出し，高精度の FP を実現する．この際，特徴量エンジニアリングを行う必要がないことが深層学習を用いる大きなメリットに挙げられる．例えば生パケットのバイト列や，パケットから得られるメタ情報（パケットサイズ，通信方向など）をそのまま深層学習の入力とすることができる．深層学習のアーキテクチャには，CNN, Autoencoder, LSTM (Long Short-Term Memory) などがよく用いられる．

以下，具体的に深層学習を用いた暗号化通信 FP の先行研究をレビューする．Aceto et al. (2019a, 2019b) は，

モバイルアプリケーションの FP に CNN, Autoencoder, LSTM を適用し，アプリケーションが受信・発信する暗号化通信からアプリケーションを特定することを試みた [13], [14]．通信種別（例：チャット，email, ストリーミングの通信等）の FP においては，Wang et al. (2017) が CNN を，Lotfollahi et al. (2020) が CNN と Autoencocer をそれぞれ適用した [15], [16]．ウェブサイト FP については多くの文献が見つかる [7], [8], [9], [10]．これらの研究は全て CNN を用いてウェブサイトの特定を行った．その際の入力は，クライアント–サーバ間の暗号化通信のうち，通信方向の系列（例：[1,-1,1,1,...]）のみを用いた．とりわけ，Sirinam et al. (2019) は，後述する triplet ネットワークを用いて高精度の CNN を学習した．Wang et al. (2020) は，スマートスピーカーとサーバ間の暗号化通信から，話者がスマートスピーカに話しかけたコマンド（例：「What is the weather today?」）を推定するボイスコマンド FP 技術を提案した [17]．モデルの学習には，CNN, Autoencder, LSTM のアンサンブル学習を用いた．

3. 関連技術

提案手法に関連する技術を説明する．

3.1 畳み込みニューラルネットワーク（CNN）

CNN は画像解析において最もよく用いられるニューラルネットワークであり，画像分類，物体検出などに用いられる．系列データ解析においても LSTM を上回る精度をあげる事例も多く，一般に系列データを扱う暗号化 FP に対しても近年最もよく用いられている．CNN を系列データに適用する場合，系列との 1 次元畳み込み演算を時間方向にスライドして実行し畳み込み結果を得る．複数の層を積み重ねてネットワークを学習することで，表現力の高い時系列特徴を抽出することが可能になる．

3.2 Triplet ネットワーク

Triplet ネットワークは triplet 損失と呼ばれる損失関数を最小化する学習法を採るニューラルネットワークである [2]．Triplet 損失は距離学習に用いられる損失であり，サンプル間の関係性（距離）を学習するために効率的な損失関数である．つまり triplet ネットワークの学習により，サンプル間の関係性を適切に表現する特徴量抽出を実現するニューラルネットワークが構成される．

以下では CNN を基にした triplet ネットワークを仮定し，記号の導入と学習方法の説明を行う．入力系列を $\boldsymbol{x} \in \mathbb{R}^{D\times T\times C}$ とする．記号 D, T, C はそれぞれ変数次元，時点数，チャネル数を表す．ある CNN モデル $f : X \to \mathbb{R}^p$ を定義する．ここで f は距離学習の文脈ではベースネットワークと呼ばれる．出力 $f(\boldsymbol{x})$ を $\boldsymbol{x}$ の特徴量と呼ぶ．

Triplet ネットワークの学習には，triplet と呼ばれる 3 つ組のサンプルセット $(\boldsymbol{x}_a, \boldsymbol{x}_p, \boldsymbol{x}_n)$ を用いる．ここで $\boldsymbol{x}_a$ はデータセットの任意のサンプルであり，アンカーサンプルと呼ばれる．サンプル $\boldsymbol{x}_p, \boldsymbol{x}_n$ はそれぞれ，ポジティブ/ネガティブサンプルと呼ばれ，アンカーサンプルと同/異クラスに属する任意のサンプルである．Triplet 損失 $L(\boldsymbol{x}_a, \boldsymbol{x}_p, \boldsymbol{x}_n)$ は，ある距離関数 $d : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ の元，以下のように定義される：

$$L(\boldsymbol{x}_a, \boldsymbol{x}_p, \boldsymbol{x}_n) = \max\left(d_{a,p} - d_{a,n} + m, 0\right). \tag{1}$$

ここで，

$$d_{a,p} = d(f(\boldsymbol{x}_a), f(\boldsymbol{x}_p)), \tag{2}$$

$$d_{a,n} = d(f(\boldsymbol{x}_a), f(\boldsymbol{x}_n)) \tag{3}$$

である．マージン $m \in \mathbb{R}_+$ はハイパーパラメータである．Triplet ネットワークの学習は，triplet の各要素を共通のベースネットワーク f に伝播し，それらの出力から定義される triplet 損失を最小化することで f の重みを最適化する．これはつまり，あるアンカーサンプルの出力（特徴量）に対し，ポジティブサンプルのそれとは距離を近く，ネガティブサンプルのそれとは距離を遠くすることを要求する．これにより f が，サンプル間の距離関係を適切に表現する特徴量を出力するように学習される．

判別問題のために triplet 損失を用いる場合は，はじめに triplet 損失の最小化によりベースネットワーク f を学習する．学習後，ベースネットワークにサンプルを入力して得られる特徴量を用いて判別モデルを学習する．このような距離学習法により，通常のクロスエントロピー損失に基づく CNN 判別モデル学習法と比較して，小標本で高精度な判別器を構成できることが知られている．

3.3 Grad-CAM

Grad-CAM は CNN を用いた分析結果に推定根拠を与える技術である [3]．画像分類において学習済み CNN を用いてサンプルのクラスを推定した際に，Grad-CAM は推定結果に対してポジティブに寄与した画像の局所領域をハイライトする．これによって，モデルがどの領域を基に推定結果を出力したかを知ることが可能になる．

以下ではタスクをクラス判別と仮定し，Grad-CAM による推定根拠の可視化を説明する．あるサンプルのクラスを推定した際に，ある出力クラス c に対する予測の寄与度を計りたいとする．そのクラスに対するスコアを y^c (softmax 関数適用前の値) と表す．ある畳み込み層 (一般に最終畳み込み層) から出力される k 番目の特徴マップを $A^k \in \mathbb{R}^{u\times v}$ とする．記号 u, v はそれぞれマップの幅・高さを表す． Grad-CAM では，特徴マップ A^k のクラス c への判別寄与度を以下のように微分に基づき算出する：

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k}. \tag{4}$$

ここで，$Z = uv$ である．その後，正方向のみに寄与した特徴マップを ReLU 関数（$\mathrm{ReLU}(x) = \max(x, 0)$）を用いて以下のように集約し，推定根拠マップ $L^c_{Grad-CAM}$ を構成する：

$$L^c_{Grad-CAM} = \mathrm{ReLU}\left(\sum_k \alpha^c_k A^k\right). \qquad (5)$$

この $L^c_{Grad-CAM}$ を入力サイズにリサイズすることで推定根拠を可視化することが可能になる．

4. 提案手法

本章では提案手法について説明する．

4.1 分析フローチャート

はじめに提案手法による暗号化通信からのリモートコマンド推定の分析フローチャートを図 1 に示す．提案システムは，リモートコマンド実行時のパケットをキャプチャし pcap ファイルを出力するキャプチャ部と，キャプチャしたパケットを分析しコマンド推定を行う分析部から成る．キャプチャ部では，操作元からリモートコマンドを実行した際に端末間で流れる通信パケットを全て取得し pcap ファイルを出力する．分析部では，はじめに pcap ファイルからリモート操作に関連するパケットを抽出する．本検証では両通信方向の SMB を含むパケットのみを抽出する．その後，暗号化 SMB メッセージに付けられたヘッダに記載された暗号化前の SMB メッセージサイズを抽出する．ここで，暗号化 SMB では SMB2 メッセージ（元の SMB ヘッダ＋ペイロード）が暗号化され，そこに暗号化のための SMB2 TRANSFORM_HEADER（平文）が新たに付与され，このヘッダ部に暗号化前の SMB メッセージサイズが記載されることに注意されたい．以上の操作により，図中央にあるような SMB メッセージサイズの系列データを構成する．この際，パディングと切り取りにより系列長を固定長 $T \in \mathbb{N}$ にする．前処理後，CNN に基づく学習済みの判別モデルにメッセージサイズ系列を入力し，コマンドのラベルを得る．

4.2 判別器の構築

続いて分析部の，SMB メッセージサイズ系列からコマンド推定を行う判別モデルの構成・学習方法について記述する．いま学習データ $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ があるとする．ここで $\boldsymbol{x}_i \in \mathbb{R}^{T\times 1}$ は前処理済みの SMB メッセージサイズ系列，$y_i \in \{1, ..., L\}$ (L: コマンド数) は対応するコマンドのラベルを表す．

4.2.1 判別モデルの構成

判別モデルには図 2 に示すような CNN を用いる．本 CNN はメッセージサイズの時系列 $\boldsymbol{x}$ を入力とし，コマンドのラベルを出力する判別器として定式化する．畳み込み層（畳み込み–バッチ正則化–活性化関数から成る）を n 層スタックし，最終畳み込み後に global average pooling を時間方向に施し，固定サイズのベクトルを得る．全結合層を 1 層挟み，出力層では softmax 関数を経てクラス毎の予測確率を得る．

4.2.2 判別モデルの学習

判別モデルの学習には，triplet 損失に基づく距離学習を採用する．はじめにベースネットワーク f を学習する．ネットワーク f に，上記に示した CNN の全結合層までを採用し，その出力 $f(\boldsymbol{x}) \in \mathbb{R}^p$ を距離を計算するための特徴量として用いる． Triplet の構成は，アンカーサンプルと，ポジティブ/ネガティブサンプルに同/異コマンド実行時のメッセージサイズ系列を選択する．ネガティブサンプルの選択には，バッチごとに最も損失が大きなサンプルを選択する方式を採用する [2]．ベースネットワーク学習後，判別器学習を行う．ベースネットワークの最終層に出力層（全結合+softmax 関数）を結合し，クロスエントロピー誤差最小化により判別器を学習する．この際，出力層以外の重みは固定し，出力層の重みのみを学習する．この結果，L 個のコマンドを判別するモデル $g : X \to \{0, 1\}^L$ が得られる．

4.3 推定根拠の可視化

上記学習済み CNN によるコマンド推定結果に対する説明性を付与するために，Grad-CAM による推定に寄与する部分系列の可視化を行う．可視化に用いる特徴マップ A^k は，最終畳み込み層の出力とし，式 (5) に従い推定根拠マップを作成する．なお本提案では，畳み込み演算時にゼロパディングを行い，かつダウンサンプリングを行わないことで，推定根拠マップをリサイズすることなく可視化することを可能にする．

4.4 関連研究との関係

本提案手法は多くの従来研究と同じく，CNN を用いた暗号化通信 FP を行う．技術的には Sirinam et al. (2019) に最も近い [9]．彼らは，ウェブサイト FP に CNN と triplet ネットワークを用いた．差異として，入力にメッセージサイズ系列を用いる点（先行研究は通信方向系列を用いる），推定根拠の可視化が可能な点が挙げられる．コマンド推定の取り組みとしては Wang et al. (2020) に近い [17]．ただし彼らの研究でのコマンドはボイスコマンドを指しており，リモートコマンドとは関係ない．また関連研究の暗号化通信 FP は総じてセキュリティ監視強化やフォレンジックへは適用されていない．本研究は知る限り，暗号化通信からのリモートコマンド推定の初の試みとなる．

5. 数値実験

暗号化通信からのコマンド推定実験の結果を掲載する．

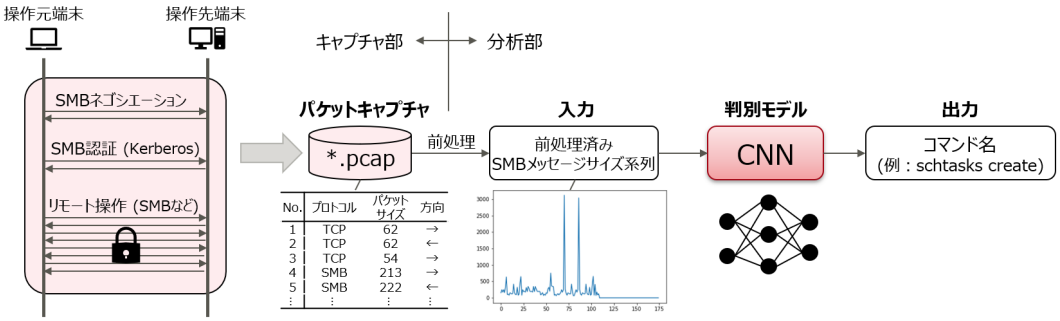


図 1　提案手法による暗号化通信からのリモートコマンド推定の流れ

Fig. 1　Flowchart of remote command inference from encrypted traffics

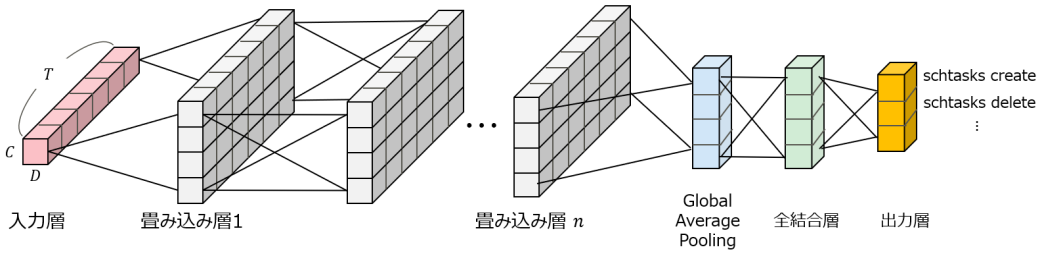


図 2　リモートコマンド判別の CNN アーキテクチャ

Fig. 2　CNN architecture for remote command classification

5.1　データセット

実験で使用するデータセットについて説明する．本実験では 36 種類 (付録 A.1) のリモート操作を対象とする．ここでコマンドの大分類・小分類の用語を定義する．大分類はコマンド (例：schtasks/sc) の分類を指し，小分類はコマンドオプション（例：schtasks create/delete/query）の分類を指す．本実験では小分類をクラス単位とした判別を行う．即ち大分類が同じで小分類が異なるリモート操作は別コマンドとして扱い，36 コマンドの判別問題を解く．

5.1.1　データセット収集

実験データとなる pcap ファイルの収集方法を説明する．実験環境の操作元 OS は Windows10 とし，操作先 OS は Windows Server 2012, 2016, 2019 の 3 通りとし，全て SMB 上でリモートプロシージャコール DCE/RPC が動作する環境を整備した．コマンド実行を操作先 OS それぞれに対し 5 回実行し pcap ファイルを取得した．このとき実行エラーが発生していないことは確認済みである．また同環境で同じように，暗号化なしの平文 pcap ファイルを取得した．このようにして 36 種類の各コマンドに対して 30 個（暗号化 15，平文 15），合計 1080 個のパケットキャプチャファイルを取得した．本実験では 1 つの pcap ファイルが 1 コマンドの実行結果となることに注意されたい．

5.1.2　データセット前処理

各 pcap ファイルに以下の前処理を行う．まずはじめに，アプリケーション層に SMB を含むパケットのみを通信方向に関係なく抽出する．暗号化 SMB メッセージに対しては，ヘッダに含まれる暗号化前の SMB メッセージサイズ情報を抽出する．平文 SMB メッセージに対してはそのままメッセージサイズを抽出する．この操作により，暗号化有無に依らず，暗号化前の SMB メッセージサイズに統一されたデータセットを構成する．

次に分析に渡す入力を構成する．各メッセージサイズ系列に対し，先頭 150 時点を抽出する．150 時点に満たない分はゼロパディングする．加えて後方 20 時点を抽出し先頭系列に結合する．相互が干渉しないよう，畳み込みフィルタサイズのゼロパディングで間を埋める．例えばフィルタサイズが 5 であれば全体の系列長は 175 となる．最後に全体最大値でデータを割って正規化する．

表 1　ハイパーパラメータ設定

Table 1　Hyperparameter settings

パラメータ	値
畳み込み層数	5
フィルタ数	128
フィルタサイズ	5
活性化関数	ReLU
全結合層ユニット数	128
オプティマイザ	Adam
学習率	0.001 (減衰あり)
バッチサイズ	64
エポック数	500

5.2　比較手法・パラメータ設定

本実験では，提案する triplet ネットワークを用いたコマ

表 2　コマンド推定実験の結果
Table 2　Performance table for command inference

	サンプル数		推定精度	
N	訓練	テスト	CNN	Triplet Net.
2	72	468	0.160 (±0.045)	0.759 (±0.052)
4	144	396	0.331 (±0.083)	0.907 (±0.024)
6	216	324	0.451 (±0.206)	0.925 (±0.029)
8	288	252	0.626 (±0.135)	0.961 (±0.008)
10	360	180	0.718 (±0.131)	0.968 (±0.005)
12	432	108	0.861 (±0.095)	**0.978 (±0.011)**

ンド推定技術を典型的な CNN と比較して検証を行う．両手法とも共通の CNN ネットワークアーキテクチャを使用する．ハイパーパラメータの設定は表 1 に示す．

5.3　実験結果

はじめに実験結果の概要を述べる．以下，推定精度は全て 5 回試行の平均値である．

はじめに，訓練サンプルが少ないケースを想定した検証を行う．ここでは 540 個の暗号化 pcap ファイルのみを用いる．36 種類の各コマンドのクラスに対し，$N(\in \{2, 4, 6, 8, 10, 12\}$ 個のサンプルを抽出し訓練データを構成し，残りをテストデータとして実験を行った．実験結果を表 2 に示す．Triplet ネットワークの推定は小標本セットでも有効な結果を示した．CNN と比較して N が小さいときのゲインは非常に大きく，$N = 4$ でも 90.7%の推定精度を得た．N が増加する毎に精度は向上し，$N = 12$ では 97.8%と非常に高い精度を記録し，CNN による推定精度を大きく上回った．また結果の分散から triplet ネットワークは安定的な学習が可能なことがわかる．CNN は結果の分散が大きく，重み初期値によって学習結果が大きく左右され，過学習防止に繊細にならなくてはならないことを示唆した．

次に，平文 pcap ファイルを学習データに追加する効果の検証を行った．上記 $N = 12$ の実験設定で，平文 pcap 540 サンプルを全て訓練データに追加して検証を行った．推定精度は 97.4%（±1.7%）となり改善は見られなかった．誤判別パターンは暗号化のみでの検証と変わらなかったため，$N = 12$ で triplet ネットワークの精度が頭打ちしたことが推察される．

続いて，同設定（暗号化のみ）で後方 20 パケットを除いて検証を行ったところ，推定精度は 95.2% (±1.2%) と悪化した．正しく判別していた psloglist read / delete コマンドが識別できなかったことが精度悪化の要因であった．

5.3.1　考察

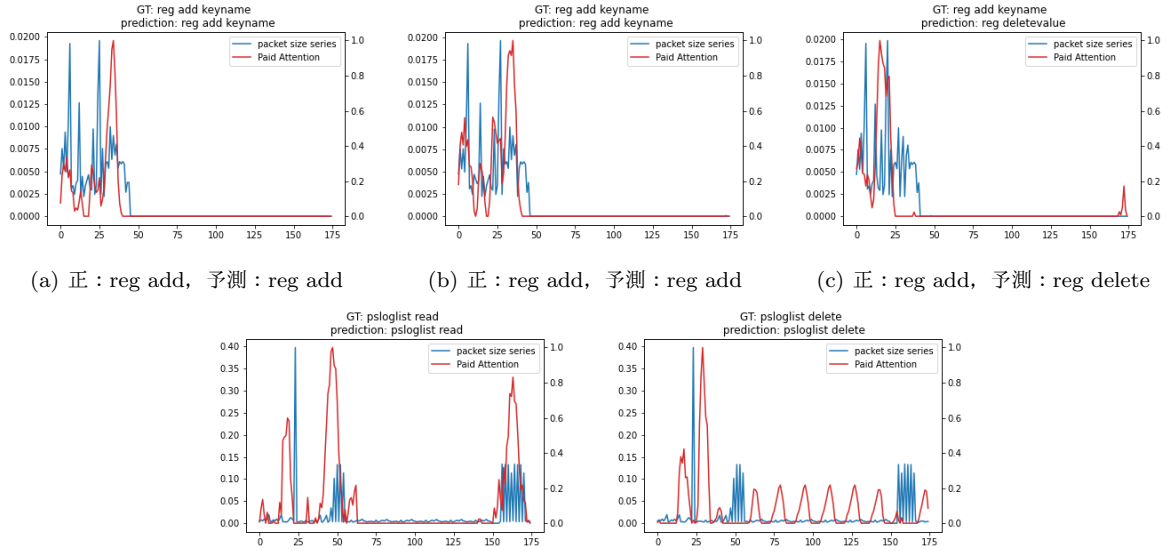
$N = 12$ 設定での実験結果を考察する．

誤判別：Triplet ネットワークでのコマンド推定では約 2%の誤判別を発生した．その内訳は全て小分類での誤判別であり，大分類の誤判別は無かった．例えば，schtasks end コマンドを schtasks query や schtasks run と誤判別するケースは存在したが，別コマンド（例：sc, psloglist）に誤判別するケースは存在しなかった．大分類と，それを更に細分化する特徴量を抽出できていることがわかる．これは triplet ネットワークが CNN とは異なり，サンプル間の距離関係を適切に学習していることを反映している．特に多かった誤判別パターンは schtasks コマンドの小分類（query, run, end），reg コマンドの小分類（add, delete）であった．これらは系列パターンが酷似しており，本提案手法による判別では判別困難であることが予想された．

後方パケットの効果：後方パケットの削除により psloglist コマンドの小分類（read, delete）が誤判別された．実際に平文 pcap ファイルを見ると，同一環境で取得された 2 つのコマンドは，前方のメッセージサイズ系列は完全に一致していた．しかし後方パケットを見ると，delete のみ ClearEventLogW を実施する SMB パケットが存在し削除処理を行っていた．このことから，後方パケットに判別に役立つ情報が存在し得ることが示唆された．また暗号化 pcap からでは，暗号化により ClearEventLogW を実施するパケットの存在を知ることはできない．しかしながら深層学習を用いることで，判別に有効な隠れた情報を手動で特徴量を構成する必要なく抽出できることがわかる．

推定根拠可視化の効果：推定根拠の可視化効果を検証する．図 3 に推定根拠の可視化例を示す．上段 (a,b) は正しく判別された reg add コマンドであり，可視化することで似た箇所の部分系列を推定根拠としていることがわかる．誤判別された reg add コマンドに対する可視化（c）は異なる部分系列を注視しており，この差が誤判別を誘発したと考えられる．下段（d,e）は 正しく判別された psloglist read/delete に対する可視化である．read に関しては前述した後方パケットを注視していることがわかる．一方 delete はその限りではい．このケースから本判別器は，判別に作用する特徴は抽出できているものの，本質的な特徴（後方パケットに潜在）を十分には注視できていない可能性がある．本実験データは同一コマンド・環境で取得した pcap ファイルが各 4 サンプルしか訓練データに含まれていないため，本質的でロバストな特徴量の抽出には課題が残ったと考えられる．しかしながら可視化が分析結果にこのような洞察を与えることができる．

特徴量可視化：CNN と比較した triplet ネットワークの効果検証のため，両手法によるコマンド特徴量を t-SNE を用いて可視化した（図 4）．CNN で抽出される特徴量と比較して，triplet ネットワークでは同クラス間の特徴量間距離が近く，異クラスとは明確な境界を引いていることがわかる．距離学習により効率よくサンプル間の関係を学習することがコマンド推定精度に寄与することが可視化からも裏付けられる．

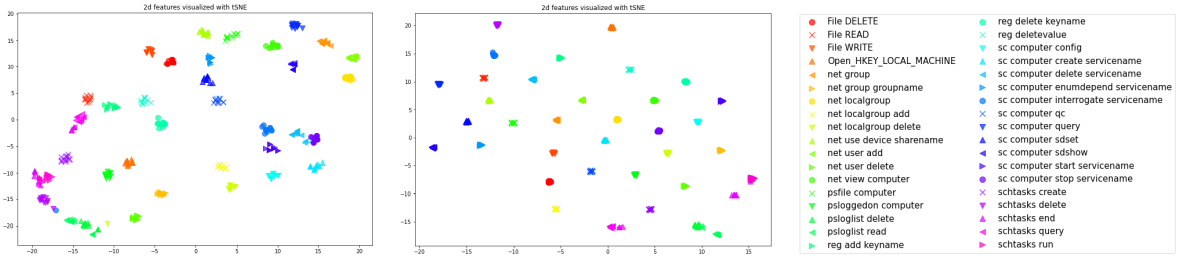


(a) 正：reg add，予測：reg add　(b) 正：reg add，予測：reg add　(c) 正：reg add，予測：reg delete

(d) psloglist read　(e) psloglist delete

図 3　可視化結果：青線はメッセージサイズ系列，赤線は Grad-CAM による推定根拠を示す

Fig. 3　Visualization of inference reason： the blue and red lines represent a packet size sequence and the reason of inference by Grad-CAM.



(a) CNN　(b) Triplet Network　(c) マーカーラベル

図 4　CNN と triplet ネットワークにより抽出される特徴量の可視化

Fig. 4　Visualization of features extracted by CNN and triplet network

6. 研究倫理

我々は情報処理学会倫理綱領に則り研究を行った．提案手法は大前提として，暗号化通信に対するフォレンジックによりセキュリティ対策に寄与することを目的とする．分析には，暗号化されない情報のみを用いる．SMB プロトコルにより秘匿化される操作先ファイルなどの引数，アカウントのクレデンシャル情報は特定できず，従って SMB プロトコルの安全性を脅かすものではない．

7. 結論

本論文では，リモートコマンド実行時の暗号化通信から実行されたコマンドを推定する深層学習技術を提案した．本技術は，コマンド実行時の pcap ファイルに含まれる SMB メッセージサイズの系列を入力，コマンド名を出力とする畳み込みニューラルネットワークを基にする．更に本技術は，triplet ネットワーク により小標本でも高精度な推定と，Grad-CAM により推定根拠の可視化を実現する．実験では，暗号化通信からのリモートコマンド推定を 97.8%の精度で実現した．これにより今後暗号化通信に対するデジタルフォレンジックを支援することが見込まれる．

今後の課題として大規模実環境での検証が求められる．実環境では更に通信バラエティ (OS などの環境依存) が多く，また大量のパケットを高速に処理することが求められるため，更なる技術検証・改良を行う必要がある．

参考文献

[1] 海野由紀，森永正信，及川孝徳，古川和快，金谷延幸，津田侑，遠峰隆史，井上大介，　鳥居悟，伊豆哲也，武仲正彦：標的型攻撃の被害範囲を迅速に分析するネットワークフォレンジック手法の提案, 暗号と情報セキュリティシンポジウム (SCIS) (2018).

[2] Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B. and Wu, Y.: Learning Fine-Grained Image Similarity with Deep Ranking, *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1386–1393 (2014).

[3] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D.: Grad-CAM: Visual Expla-

nations from Deep Networks via Gradient-Based Localization, *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626 (2017).

[4] Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A. and Tarkoma, S.: IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT, *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 2177–2184 (2017).

[5] Panchenko, A., Lanze, F., Pennekamp, J., Engel, T., Zinnen, A., Henze, M. and Wehrle, K.: Website Fingerprinting at Internet Scale., *NDSS* (2016).

[6] Hayes, J. and Danezis, G.: k-fingerprinting: A robust scalable website fingerprinting technique, *25th USENIX Security Symposium*, pp. 1187–1203 (2016).

[7] Rimmer, V., Preuveneers, D., Juarez, M., Van Goethem, T. and Joosen, W.: Automated Website Fingerprinting through Deep Learning, *Proceedings of the 25nd Network and Distributed System Security Symposium (NDSS 2018)*, Internet Society (2018).

[8] Sirinam, P., Imani, M., Juarez, M. and Wright, M.: Deep fingerprinting: Undermining website fingerprinting defenses with deep learning, *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1928–1943 (2018).

[9] Sirinam, P., Mathews, N., Rahman, M. S. and Wright, M.: Triplet Fingerprinting: More Practical and Portable Website Fingerprinting with N-shot Learning, *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1131–1148 (2019).

[10] Bhat, S., Lu, D., Kwon, A. and Devadas, S.: Var-CNN: A data-efficient website fingerprinting attack based on deep learning, *Proceedings on Privacy Enhancing Technologies*, Vol. 2019, No. 4, pp. 292–310 (2019).

[11] Oh, S. E., Sunkam, S. and Hopper, N.: p1-FP: Extraction, Classification, and Prediction of Website Fingerprints with Deep Learning, *Proceedings on Privacy Enhancing Technologies*, Vol. 2019, No. 3, pp. 191–209 (2019).

[12] Taylor, V. F., Spolaor, R., Conti, M. and Martinovic, I.: Appscanner: Automatic fingerprinting of smartphone apps from encrypted network traffic, *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE, pp. 439–454 (2016).

[13] Aceto, G., Ciuonzo, D., Montieri, A. and Pescapé, A.: Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges, *IEEE Transactions on Network and Service Management*, Vol. 16, No. 2, pp. 445–458 (2019a).

[14] Aceto, G., Ciuonzo, D., Montieri, A. and Pescapè, A.: MIMETIC: Mobile encrypted traffic classification using multimodal deep learning, *Computer Networks*, Vol. 165, p. 106944 (2019b).

[15] Wang, W., Zhu, M., Wang, J., Zeng, X. and Yang, Z.: End-to-end encrypted traffic classification with one-dimensional convolution neural networks, *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, IEEE, pp. 43–48 (2017).

[16] Lotfollahi, M., Siavoshani, M. J., Zade, R. S. H. and Saberian, M.: Deep packet: A novel approach for encrypted traffic classification using deep learning, *Soft Computing*, Vol. 24, No. 3, pp. 1999–2012 (2020).

[17] Wang, C., Kennedy, S., Li, H., Hudson, K., Atluri, G., Wei, X., Sun, W. and Wang, B.: Fingerprinting Encrypted Voice Traffic on Smart Speakers with Deep Learning, *Proceedings of 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 254–265 (2020).

付　　録

A.1 実験対象とした 36 コマンドリスト

表 A·1 実験対象のリモートコマンドリスト（※コマンド名は識別用の名称であり実際に端末で実行するコマンドではない）

Table A·1 List of remote comannds subject to experiments

id	大分類	コマンド名
1	ファイル操作	File DELETE
2		File READ
3		File WRITE
4	レジストリルートキー参照	Open_HKEY_LOCAL_MACHINE
5	グローバルグループ操作	net group
6		net group groupname
7	ローカルグループ操作	net localgroup
8		net localgroup add
9		net localgroup delete
10	ドライブ割り当て	net use device sharename
11	アカウント操作	net user add
12		net user delete
13	コンピューター覧参照	net view computer
14	リモートファイル管理	psfile computer
15	ログオンユーザ管理	psloggedon computer
16	イベントログ操作	psloglist delete
17		psloglist read
18	レジストリ操作	reg add keyname
19		reg delete keyname
20		reg deletevalue
21	Windows サービス操作	sc computer config
22		sc computer create servicename
23		sc computer delete servicename
24		sc computer enumdepend servicename
25		sc computer interrogate servicename
26		sc computer qc
27		sc computer query
28		sc computer sdset
29		sc computer sdshow
30		sc computer start servicename
31		sc computer stop servicename
32	タスクスケジューラ操作	schtasks create
33		schtasks delete
34		schtasks end
35		schtasks query
36		schtasks run