Adversarial trainingを用いた未知ファミリ検知手法

芝原 俊樹[1,2,a)]　千葉 大紀[1]　秋山 満昭[1]　波戸 邦夫[1]　村田 正幸[2]

概要：攻撃者は，機械学習を応用した検知手法を回避するために新たな攻撃ツールを開発し続けている．この問題に対応するためには，教師データに含まれない新たな攻撃ツールで作成された悪性データ（未知ファミリ）の検知が重要である．本稿では，攻撃の成立に必要となる基盤的な特徴が，複数のファミリに横断的に内在することに着目し，類似する既知ファミリに共有の特徴に基づく分類手法を提案する．そのため，提案手法の学習には，domain adaptation に用いられている adversarial training を応用する．Domain adaptation は，もともと写真とイラスト等の特性の異なるデータセットを用いて，物体の形状や色等のデータセットに共有の特徴に基づく分類を実現するものであるが，提案手法では，複数のファミリに共有の特徴である悪用されるアプリケーションや厳格でないレジストラから取得されたドメイン名に基づく分類に適用する．これらの特徴は，攻撃者が自由に設定できないため，未知ファミリの検知にも有効であると考えられる．実際，悪性通信系列検知システムに提案手法を適用した評価では，一般的なニューラルネットワークと比較して，教師データに含まれないファミリの検知率を最大 19%向上させられることを示した．

キーワード：domain adaptation，deep learning，drive-by download 攻撃

1. はじめに

研究者やセキュリティベンダは，マルウェアや悪性サイト等の悪性データの検知に機械学習を応用することで，既知悪性データとの類似度に基づいて，新規悪性データを検知する手法を提案している [1–3]．一方で，攻撃者はこれらの検知手法を回避するために，新たな攻撃ツールを開発し，これまでとは特徴の異なる悪性データを作成している [4]．日々進化する攻撃ツールで作成された悪性データによる被害を防止するためには，従来手法で想定されていた教師データに含まれている攻撃ツールで作成された悪性データ（既知ファミリ）の検知だけでなく，教師データに含まれていない攻撃ツールで作成された悪性データ（未知ファミリ）の検知も重要である．

未知ファミリの特徴と既知ファミリの特徴が異なるほど，機械学習を応用した手法での未知ファミリの検知は困難となる．しかし，攻撃の効率や成功率に影響するため，攻撃者が自由に設定できない特徴も存在する．例えば，drive-by download 攻撃で悪用されるアプリケーションは，Flash，PDF，Java 等に限られており，悪用するドメイン名も管理が厳格ではないレジストラから取得されることが多い [5]．攻撃を成立させるために必要となる基盤的な特徴は，異なる複数の既知ファミリや未知ファミリに横断的に内在すると考えられる．このため，本稿では，未知ファミリの検知性能を向上させることを目的として，複数の既知ファミリに共有の特徴に基づく分類器を構築する．

このような分類器の構築には，相異なる最適化の両立と，検知性能向上の安定化という 2 つの課題が存在する．1 つ目の課題の相異なる最適化とは，特徴の共有性の最適化と，良性/悪性の分類精度の最適化である．特徴の共有性のみを最適化して，多くの既知ファミリに共有の特徴のみに基づく分類器を構築すると，誤検知が増加して良性/悪性の分類精度が低下する．このため，両者を同時に最適化する必要がある．2 つ目の課題の検知性能向上の安定化とは，特徴が大きく異なるファミリを含むデータセットを利用する際に，未知ファミリ検知の性能を安定して向上させることである．例えば，攻撃に Flash を悪用するファミリと，PDF を悪用するファミリに共有の特徴には，未知ファミリ検知に有効な悪用するアプリケーションの特徴が含まれていない．このため，未知ファミリの検知性能を安定して向上させるためには，ファミリごとの類似度を考慮する必要がある．

1 つ目の課題を解決するために，我々は，ファミリに共有の特徴に基づく分類と，domain adaptation [6,7] で実現されている特性の異なるデータセットに共有の特徴に基づく分類が類似していることに着目し，domain adaptation

[1] NTT セキュアプラットフォーム研究所
[2] 大阪大学
[a)] toshiki.shibahara.de@hco.ntt.co.jp

表 1　良性/悪性通信の例

ラベル	ファミリ	URL	Content-type
良性		www.ntt.co.jp/news2018/1807/180718a.html	html
悪性	Rig	[snipped].northwestfloridacannabis.org/?ct=kulture&oq=X96cpLOFRaAG[snipped]	x-shockwave-flash
	Neutrino	[snipped].morgansdecorators.com/street/[snipped]Y3B5eA.swf	x-shockwave-flash
	Magnitude	[snipped].dropsfry.gdn/d9947c8e03e9dc40167c02718275b280?win[snipped]	x-shockwave-flash

の学習で用いられている adversarial training を適用する．Adversarial training は，複数のニューラルネットワークを敵対させる最適化手法であり，データセットに共有の特徴空間の最適化と，分類精度の最適化を同時に実施している．例えば，写真とイラストのデータセットからは，写真に固有のテクスチャや陰影等の特徴ではなく，データセットに共有かつ分類に有効な，物体の形状や色に基づく分類器を構築している [7]．そこで，本稿では，adversarial training を応用することで，ファミリに共有の特徴空間の最適化と，分類精度の最適化を両立させる．具体的には，ニューラルネットワークを用いた良性/悪性の分類の最適化と同時に，ニューラルネットワークの中間層の値（representation）に基づいて既知ファミリを分類した際に，ファミリの分類が失敗するように representation を更新する．Representation にファミリ固有の特徴が含まれているとファミリの分類が成功するため，ファミリの分類が失敗するように更新することで，ファミリに共有の特徴を表象する representation を得ることができる．

2 つ目の課題を解決するために，我々は，類似度の高いファミリに共有の特徴に基づく分類器を構築する．そのために，事前に既知ファミリの分類を実施し，そのときの予測確率が上位のファミリに共有の特徴を分類に用いる．分類が困難なファミリには共有の特徴が多いため，未知ファミリの検知に有効な特徴が含まれている可能性が高いと考えられる．具体的には，representation に基づくファミリの分類結果が，事前に既知ファミリを分類した際の予測確率を平滑化したものに近づくように更新する．

上記の通り，本稿では，未知ファミリの検知性能を向上させるために，adversarial training を用いて類似のファミリに共有の特徴に基づいて分類する手法を提案する．我々は，提案手法を悪性通信系列検知システムに適用し，未知ファミリの検知における有効性を評価した．我々の主な貢献は下記のとおりである．

- Domain adaptation で用いられている adversarial training を初めてサイバーセキュリティに応用し，異なるファミリに共通の特徴に基づく分類器の構築に有効であることを示した．
- 類似のファミリに共通の特徴に基づく分類器を構築することで，未知ファミリの検知性能を向上させられることを示した．

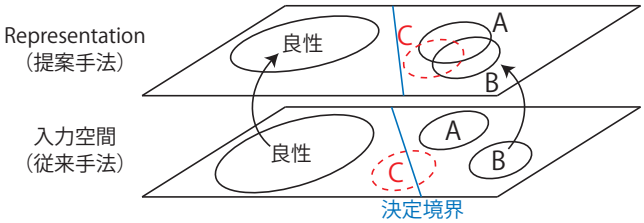

図 1　提案手法と従来手法の比較．ファミリ A および B は既知，C は未知ファミリを想定．

2. 研究背景

本章では，提案手法の効果，数式の表記，提案手法に応用する adversarial training について説明する．

2.1 Motivating Example

具体的な良性/悪性通信の例を用いて，提案手法で期待される効果を説明する．良性通信と 3 つのファミリの悪性通信の例を表 1 に示す．悪性通信は，drive-by download 攻撃で悪用されている悪性サイトへの通信である．同じ攻撃に悪用される悪性 URL であっても，URL の構造はファミリごとに大きく異なっている．例えば，Rig と Magnitude の URL にはクエリ文字列が含まれているが，Neutrino には含まれていない．同様に，ドメイン名の長さや URL のパスの階層数もファミリごとに異なっている．一方で，すべてのファミリに共通の特徴も存在する．例えば，content-type の種別や URL にランダムな文字列が使われていることは，すべてのファミリに共通している．

一般的な機械学習手法と提案手法の違いを図 1 を用いて説明する．一般的な機械学習手法では，通信の特徴が入力されると，入力空間で分類精度が高くなるような決定境界を求める．表 1 のようにファミリごとの特徴が異なる場合，それぞれのファミリは入力空間では異なる座標に位置している．このため，図 1 において，ファミリ A と B が既知，ファミリ C が未知であった場合，ファミリ A および B と良性データが分類できるような決定境界では，ファミリ C の検知率は低くなってしまう．

提案手法では，類似のファミリの分類が失敗する representation に基づいて分類を実施する．ファミリ A または B に固有の特徴が representation に伝搬されていると分類が成功するため，分類が失敗する representation には，ファミリ A と B に共通の特徴が伝搬されるようになる．この

ような representation では，ファミリ A と B は近い座標に位置するようになる．さらに，表 1 のすべてのファミリに共通の特徴のように，ファミリ A と B に共通の特徴がファミリ C にも共通していた場合，ファミリ C も近い座標に位置する．この representation でファミリ A および B と良性データが分類できる決定境界を求めると，ファミリ C も検知可能となる．

2.2 Notation

本稿での数式の表記方法について説明する．学習やテストに用いるデータセットは $\mathbf{X} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ と表記する．ここで，$\mathbf{x}_i \in \mathbb{R}^{N_x}$ は入力データ，$\mathbf{y}_i \in \mathbb{R}^{N_y}$ はラベルであり，N_x は入力データの次元数，N_y はクラス数である．ラベルは one-hot representation を用いて N_y 次元のベクトルとして表現する．つまり，$\mathbf{x}_i$ に対応するクラスが k であった場合，k 次元目の値のみ 1 で他は 0 となるベクトルであり，$y_{ij} = 1\ (j = k),\ y_{ij} = 0\ (j \neq k)$ となる．

本稿では，複数の層からなるニューラルネットワークを関数の形式で表記する．ニューラルネットワーク F に $\mathbf{x}$ を入力した場合，予測されたラベルを $\hat{\mathbf{y}} = F(\mathbf{x}; \boldsymbol{\theta}_F)$ と表記する．ここで，$\boldsymbol{\theta}_F$ はニューラルネットワークのパラメータである．ニューラルネットワークをある層の前後で 2 つの部分に明示的に分けて考える際には，$\hat{\mathbf{y}} = F_1(F_0(\mathbf{x}; \boldsymbol{\theta}_{F_0}); \boldsymbol{\theta}_{F_1})$ と書く．入力 $\mathbf{x}$ を与えた際のニューラルネットワークの中間層の値を representation と呼び，$\mathbf{h}$ で表記する．例えば，F_0 を適用した後の representation は $\mathbf{h} = F_0(\mathbf{x}; \boldsymbol{\theta}_{F_0})$ となる．以後，簡単のため，$F(\mathbf{x}; \boldsymbol{\theta}_F)$ と $F(\mathbf{x})$ を同じ意味で使用する．

2.3 Adversarial Training

提案手法の基礎となる技術である domain adaptation [6, 7] における adversarial training について一般的な適用例を説明する．Domain adaptation ではラベルつきデータセット $\mathbf{X}_l = \{(\mathbf{x}_i^l, \mathbf{y}_i^l)\}_{i=1}^{N_l}$ とラベルなしデータセット $\mathbf{X}_u = \{\mathbf{x}_i^u\}_{i=1}^{N_u}$ を考える．ここでの目的は，ラベルつきデータセットを活用して，ラベルなしデータセットを高精度に分類できる分類器を構築することである．ラベルつきデータセットとしては，ラベルつきデータの入手が容易な写真のデータセット，ラベルなしデータセットとしては，ラベルつきデータの入手が比較的困難なイラスト等のデータが用いられる [7]．以後，簡単のため，ラベルつきデータセットを source，ラベルなしデータセットを target とよぶ．

図 2 に示す入力に近い層は共有し，その後は 2 つの部分に分岐するニューラルネットワークを考える．分岐した後のネットワークでは，クラスラベルの分類と入力データが source と target のどちらに属するかの分類を実施する．共有されたネットワークを F_s，クラスラベルの分類を実施するネットワークを F_c，データセットの分類を実施するネットワークを F_d とすると，入力 $\mathbf{x}$ に対するクラスの予測 $\hat{\mathbf{y}}$ とデータセットの予測 $\hat{\mathbf{d}}$ は，$\hat{\mathbf{y}} = F_c(F_s(\mathbf{x}))$，$\hat{\mathbf{d}} = F_d(F_s(\mathbf{x}))$ となる．ここで，$\hat{\mathbf{d}} \in \mathbb{R}^2$ であり，データが source の場合は $\mathbf{d} = [1, 0]$，target の場合は，$\mathbf{d} = [0, 1]$ である．

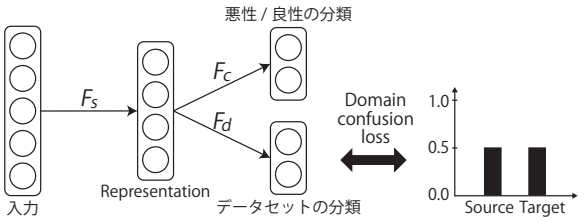

図 2 Domain adaptation 手法概要

Domain adaptation では，source と target に共通の representation を活用して，ラベルなしデータのための分類器を構築する．具体的には，共有されたネットワーク F_s を適用した後の representation ($\mathbf{h} = F_s(\mathbf{x})$) が source と target で類似するように学習を実施する．このために，データセットの分類が成功するように F_d の学習を実施し，それと同時にデータセットの分類が失敗するように F_s の学習を実施する．この結果，source と target で異なる特徴は F_s を伝播せず，共通の特徴のみが伝播するようになる．この representation に基づいて，source のラベルを用いて構築された分類器は，target にも共通の特徴に基づいて分類を実施するため，target のクラス分類にも適用することができる．Source が写真で，target がイラストであった場合には，representation はデータセットに共通する物体の形状や色を表象する空間となることが想定される．形状や色の特徴は写真でもイラストでも類似しているため，この representation を用いて写真の分類が成功すれば，同様にイラストの分類も成功する [7]．

Adversarial training では，3 つの損失を考える．1 つ目は，クラス分類の損失であり，classification loss と呼ぶ．一般的にはクロスエントロピーを用いて下記を考える．

$$L_c = -\sum_{i=1}^{N_l}\sum_{j=1}^{N_y} y_{ij}^l \log \hat{y}_{ij}^l \tag{1}$$

2 つ目は，データセットの分類の損失であり，domain prediction loss と呼ぶ．同様にクロスエントロピーを考える．

$$L_d = -\sum_{i=1}^{N_l+N_u}\sum_{j=1}^{2} d_{ij} \log \hat{d}_{ij} \tag{2}$$

3 つ目は，データセットの分類の失敗度合いを意味する損失であり，domain confusion loss と呼ぶ．最もデータセットを区別できていない状態は，データセットの予測確率が等確率になる場合である．そこで，予測確率と均一分布のクロスエントロピーで損失を定義する．

$$L_{conf} = -\sum_{i=1}^{N_l+N_u}\sum_{j=1}^{2} \frac{1}{2} \log \hat{d}_{ij} \tag{3}$$

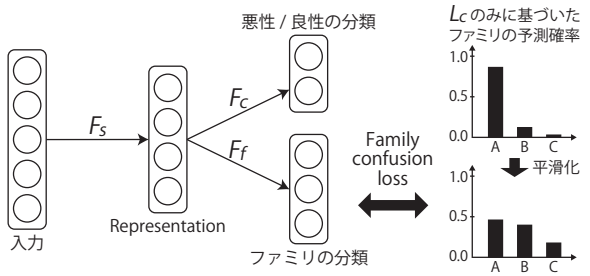

図 3 提案手法概要

最適化では，$\boldsymbol{\theta}_s, \boldsymbol{\theta}_c$ に関する classification loss と domain confusion loss の最小化と，$\boldsymbol{\theta}_d$ に関する domain prediction loss の最小化を交互に実施する．

$$\min_{\boldsymbol{\theta}_s, \boldsymbol{\theta}_c} \quad L_c + L_{conf} \tag{4}$$

$$\min_{\boldsymbol{\theta}_d} \quad L_d \tag{5}$$

このように学習を実施することで，クラス分類は成功するが，データセットの分類が失敗するような representation の学習を実現できる．このような学習手法は，複数のネットワークを敵対させながら学習させるため，adversarial training と呼ばれている [6]．

3. 提案手法

類似するファミリに共通の特徴に基づいて分類する提案手法の詳細を説明する．学習には，データセット $\mathbf{X} = \{(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)\}_{i=1}^N$ を用いる．ここで，$\mathbf{x}_i$ は入力データ，$\mathbf{y}_i \in \mathbb{R}^2$ はラベル，$\mathbf{z}_i \in \mathbb{R}^{N_z}$ はファミリラベルである．ここで，$\mathbf{y}_i = [1, 0]$ は良性を意味し，$\mathbf{y}_i = [0, 1]$ は悪性を意味する．N_z はファミリの数である．ファミリラベルはデータが悪性の場合のみに付与される．

図 3 に示す途中で分岐のあるニューラルネットワークを使用して分類器を構築する．分岐後のネットワークでは，良性/悪性のクラス分類と，悪性データのファミリ分類を実施する．共有のネットワークを F_s，クラス分類を実施するネットワークを F_c，ファミリ分類を実施するネットワークを F_f と書く．

3.1 学習方法

Domain adaptation と同様に，3 つの損失を adversarial training を用いて最適化する．1 つ目はクラス分類に関する損失であり，クロスエントロピーを用いて下記のように定義する．

$$L_c = -\sum_{i=1}^{N}\sum_{j=1}^{2} y_{ij} \log \hat{y}_{ij} \tag{6}$$

2 つ目は，ファミリ分類の損失であり，family prediction loss と呼ぶ．同様にクロスエントロピーで定義する．

$$L_f = -\sum_{i=1}^{N} (\mathbb{1}[y_{i2} = 1] \sum_{j=1}^{N_z} z_{ij} \log \hat{z}_{ij}) \tag{7}$$

ファミリラベルは悪性データにしか付与されていないため，クラスラベルが悪性の場合のみに family prediction loss を算出する．3 つ目はファミリの分類の失敗度合いを表す family confusion loss であり，L_{conf} と書く．定義は次節で説明する．最適化では，$\boldsymbol{\theta}_s, \boldsymbol{\theta}_c$ に関する classification loss と family confusion loss の最小化と，$\boldsymbol{\theta}_f$ に関する family prediction loss の最小化を交互に実施する．

$$\min_{\boldsymbol{\theta}_s, \boldsymbol{\theta}_c} \quad L_c + L_{conf} \tag{8}$$

$$\min_{\boldsymbol{\theta}_f} \quad L_f \tag{9}$$

3.2 Family Confusion Loss

類似したファミリに共通する representation の学習のために，類似したファミリの分類が失敗するように representation を更新する．このとき，どのファミリが類似しているかを特定するために，$\boldsymbol{\theta}_s, \boldsymbol{\theta}_c$ に関する L_c の最小化のみを実施した場合のファミリの予測確率を利用する．この予測確率を平滑化したものと，representation に基づくファミリの予測確率のクロスエントロピーで family confusion loss を定義する．

L_c の最小化のみを実施した際の $\mathbf{x}_i$ に対するファミリ j の予測確率を p_{ij} とし，平滑化後の予測確率 p'_{ij} を下記のように定義する．

$$p'_{ij} = \frac{p_{ij} + a}{\sum_{k=1}^{N_z}(p_{ik} + a)} \tag{10}$$

ここで，$a \in \mathbb{R}$ は定数である．この p'_{ij} を用いて family confusion loss を下記のように定義する．

$$L_{conf} = -\sum_{i=1}^{N} (\mathbb{1}[y_{i2} = 1] \sum_{j=1}^{N_z} p'_{ij} \log \hat{z}_{ij}) \tag{11}$$

family confusion loss も family prediction loss と同様に，悪性の場合にのみ損失を算出する．類似するファミリほど p'_{ij} の値は近くなっているため，L_{conf} を最小化することで，類似のファミリの分類が失敗する representation を得ることができる．

4. 実験設定

提案手法の有効性を評価するために，悪性通信系列検知システムを構築し，提案手法を適用した際の未知ファミリの検知性能を評価する．本章では，構築した悪性通信系列検知システム，比較手法，データセット，ハイパーパラメータ最適化について説明する．

4.1 悪性通信系列検知システム

本システムでは，プロキシログを入力として受け付け，exploit kit によって作成された drive-by download 攻撃に悪用される悪性サイトへの通信かどうかの分類結果を出力する．プロキシログには，各端末がインターネットに接続

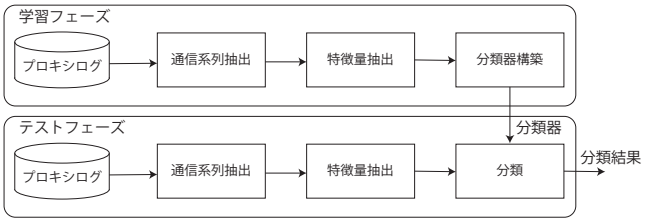

図 4 悪性通信系列検知システム

する際の HTTP リクエストが記録されている．ログには，時刻，送信元 IP アドレス，宛先 URL，HTTP メソッド，HTTP ヘッダが含まれている．Exploit kit で作成された悪性サイトへの通信が複数回発生する場合は，複数の通信の順序関係を考慮して分類することで個々の通信のみに基づく場合よりも高い分類性能を達成可能である [8]．良性サイトへの通信も同一サイトへの通信が複数回発生するため，これらを考慮した方が悪性サイトとの分類が容易となる．同一ドメイン名への通信であってもホスティングサービスを利用している場合等は，異なるサイトへの通信である可能性もあるため，本システムでは同一端末から同一宛先 FQDN への通信を抽出して系列を作成し，その系列から特徴量を抽出する．

システムの構成を図 4 に示す．学習フェーズでは，ラベルとファミリが既知のプロキシログから通信の系列を抽出し，各系列から特徴量を抽出する．その後，これらの特徴量を使用して分類器の学習を実施する．テストフェーズでは，ラベルが未知のプロキシログを入力し，通信系列の抽出と特徴量抽出を実施する．その後，学習フェーズに構築した分類器で分類を実施する．以降で通信系列抽出，特徴量抽出，分類器について詳細に説明する．

通信系列抽出 通信系列としてプロキシログから，同一送信元 IP アドレスおよび同一の宛先 FQDN ごとの通信の系列を抽出する．系列が長くなりすぎるのを防ぐため，系列の最初の通信から 2 分以上経過した場合，または通信数が 100 を超えた場合に系列を分割する．分割後の通信から新たな系列の作成を開始する．

特徴量抽出 従来の drive-by download 攻撃検知手法 [3,5] を参考に特徴量を設計した（表 2）．系列に含まれる個々の通信から特徴量を抽出した後に，複数の通信の特徴量を統合する．使用した特徴量は，URL 以外の部分から抽出した一般的な特徴量と，URL に関する特徴量の 2 つに大別される．一般的な特徴量は，通信間隔，系列内での同一通信の有無，HTTP メソッドと content-type の組み合わせである．URL に関する特徴量は，IP アドレスによる宛先指定，サブドメインの有無，TLD の人気度，FQDN/パス/ファイル名/クエリ/URL の長さ，パスの階層数，バイナリファイルの有無，拡張子が.php または.html，拡張子が.js，拡張子が.pdf または.swf，クエリの数，クエリ中の大文字/小文字/数字/記号の割合である．HTTP メソッドは，GET，POST，その他，content-type は text，application，binary，image，video，audio，font，multipart，その他，から当てはまるものを選択する．このため，HTTP メソッドと content-type の組み合わせは 27 通りとなる．TLD の人気度は，同一 TLD のドメイン名の中で最も高い Alexa[*1]の順位を用いる．

表 2 特徴量

分類	番号	特徴量	形式
一般	1	通信間隔	実数
	2	系列内での同一の通信の有無	2 値
	3	HTTP メソッドと content-type	カテゴリ
URL	4	IP アドレスによる宛先の指定	2 値
	5	サブドメインの有無	2 値
	6	TLD の人気度	実数
	7	FQDN の長さ	実数
	8	パスの長さ	実数
	9	ファイル名の長さ	実数
	10	クエリの長さ	実数
	11	URL の長さ	実数
	12	パスの階層数	実数
	13	バイナリファイルの有無	2 値
	14	拡張子が.php または.html	2 値
	15	拡張子が.js	2 値
	16	拡張子が.pdf または.swf	2 値
	17	クエリの数	実数
	18	クエリ中の大文字の割合	実数
	19	クエリ中の小文字の割合	実数
	20	クエリ中の数字の割合	実数
	21	クエリ中の記号の割合	実数

系列の特徴量として個々の通信の特徴量を統合する方法は，特徴量の形式ごとに異なる方法で実施する．実数の場合は，特徴量の平均と標準偏差，2 値の場合は合計と平均，カテゴリの場合は 1-gram と 2-gram を用いる．さらに，ニューラルネットワークに入力する際には，特徴量ごとのスケールを合わせる必要があるため，各特徴量の平均が 0 分散が 1 となるように標準化を実施する．最終的な特徴ベクトルの次元数は 793 となる．

分類器 提案手法を適用するニューラルネットワークの構成を説明する．F_s，F_c，F_f ともに 2 層の全結合ネットワークを採用し，F_c と F_f の最終層の活性化関数は softmax，それ以外は ReLU を用いる．さらに，F_c には過学習を防ぐために dropout [9] を適用する．学習フェーズの最適化には adam [10] を適用する．

4.2 比較手法

提案手法の効果を評価するために，2 つの手法と比較する．1 つ目は，一般的なニューラルネットワーク（baseline）

[*1] https://www.alexa.com/topsites

である．この手法は，4.1 節に記載のネットワークの構造は変えずに L_c のみを最小化するように学習させた場合に相当する．2 つ目は，ファミリごとの類似度を考慮せずに，すべてのファミリに共通の representation を学習させる手法（equal confusion）である．この手法は，domain adaptation と同様に family confusion loss を均一分布とのクロスエントロピーで定義した場合に相当する．つまり，$L_{conf} = -\sum_{i=1}^{N}(\mathbb{1}[y_{i1}=1]\sum_{j=1}^{N_z}\frac{1}{N_z}\log\hat{z}_{ij})$ となる．

4.3 データセット

実験に用いるデータセットについて説明する．悪性データは Malware traffic analysis [*2] または Broad analysis [*3] から収集した pcap をプロキシログに変換して用意した．ファミリは，Rig，Neutrino，Magnitude，Sundown が含まれている．良性データには企業のプロキシログを用いた．このログはユーザの同意に基づいて収集されたものである．さらに，プライバシーに配慮し，企業および個人を特定できる情報は記録されないように収集を実施した．

悪性データは収集された日に基づいて 2 分割し，収集された日が早いものを教師データ，遅いものをテストデータとして使用した．良性データは 2017/10/10 に収集したデータを教師データ，2018/1/16 に収集したデータをテストデータとして用いた．全データの系列数の合計は，110,856 であり，ファミリごとの収集期間と系列数を表 3 に示す．

提案手法によって未知ファミリの検知性能を向上させることができるかを評価するために，悪性ファミリのうち 1 つを未知と想定したデータセットを 4 通り作成し評価を実施した．例えば，Rig を未知と想定した場合は，教師データのうち Rig 以外の悪性データと良性データを学習に使用し，テストデータのうち Rig の悪性データと良性データを評価に使用する．以後，教師/テストデータと記載した場合は，悪性ファミリのうち 1 つを未知と想定したデータセットの教師/テストデータを指す．

4.4 ハイパーパラメータ最適化

ニューラルネットワークのハイパーパラメータを最適化するために，教師データを使用したクロスバリデーションを行った．教師データには 3 つのファミリが含まれているため，そのうち 1 つを評価用データ，残りを学習用データとした評価を 3 通り行った．このとき，良性データはランダムに 2 分割して，学習用と評価用として用いた．この結果，最も分類性能が高かったハイパーパラメータを選択した．分類性能の指標としては，pAUC を使用した．pAUC とは，誤検知率が閾値以下の Receiver Operating Characteristic（ROC）曲線の下側の面積である．本稿では，サイバーセキュリティでは，高くても誤検知率を数パーセントに設定することから，閾値を 0.1 とした．

*2 https://www.malware-traffic-analysis.net/
*3 http://www.broadanalysis.com/

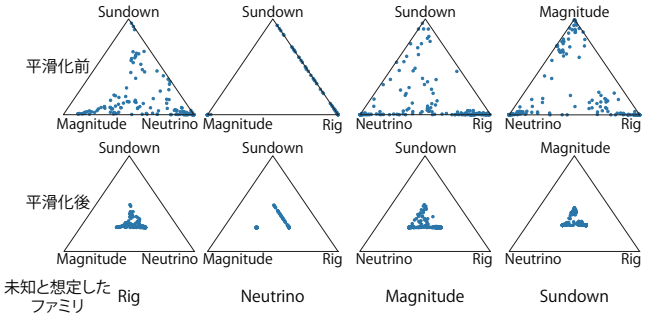

図 5 平滑化前後の予測確率

クロスバリデーションの結果，ニューラルネットワークのハイパーパラメータとして中間層のニューロン数は 10，式 (8) の学習係数は 0.001，式 (9) の学習係数は 0.00001，dropout の割合は 0.1，バッチサイズは 100，エポック数は 500 を選択した．提案手法の平滑化のハイパーパラメータ a は，Sundown を未知と想定したデータセットでは 1.5，それ以外では 1.0 を選択した．

5. 実験結果

本章では，提案手法の評価結果を説明する．さらに，提案手法と比較手法との違いを明確にするために，平滑化による予測確率の変化と処理時間についても評価した．

5.1 平滑化による予測確率の変化

提案手法によって適切に予測確率の平滑化が実施できているかを評価した．平滑化前後の各データセットでのファミリの予測確率を図 5 に示す．図中の各点はデータを意味し，その位置が予測確率を示している．ファミリ名が記載されている各頂点に点が位置している場合は，そのファミリの予測確率が 1，その他が 0 である．三角形の中心に点が位置している場合は，3 つのファミリの予測確率が等確率であることを示している．想定通りに，平滑化によってどのファミリ同士が類似しているかの傾向は保持したまま，均一分布に近づいていることが分かる．さらに，ファミリごとに類似度が異なっているため，どのファミリが類似しているかを考慮することは効果があると推測される．

5.2 未知ファミリの検知性能

未知ファミリの検知性能を評価するために，pAUC（誤検知率の閾値 0.1）と ROC 曲線を用いた．各手法の pAUC を表 4 に示す．参考として，すべてのファミリを教師データとして使用した場合の検知性能（oracle）も記載してある．Neutrino を未知と想定した場合は，すべての手法で oracle と同等の検知性能が達成できていた．その他のファミリを未知と想定した場合は，oracle より検知性能が低下していた．これらのファミリでは，提案手法の方が比較手

表 3　データセット

ラベル	ファミリ	教師 期間	教師 系列数	テスト 期間	テスト 系列数
良性		2017/10/10	10,000	2018/1/16	100,000
悪性	Rig	2015/5/7–2016/11/5	270	2016/11/7–2017/10/25	270
	Neutrino	2013/6/19–2016/7/12	97	2016/7/13–2016/9/26	97
	Magnitude	2014/1/15–2015/3/28	41	2015/3/28–2017/8/5	42
	Sundown	2015/12/27–2016/12/29	19	2016/12/29–2017/3/7	20

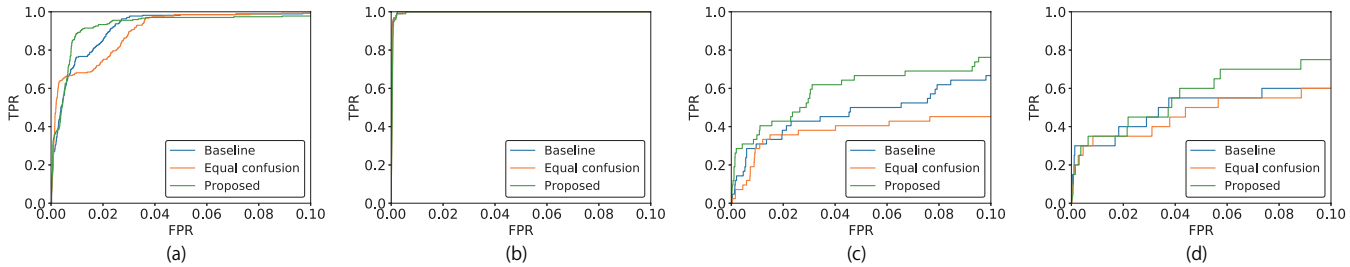

図 6　未知ファミリ検知における ROC 曲線．未知と想定したファミリはそれぞれ (a) Rig, (b) Neutrino，(c) Magnitude，(d) Sundown である．

表 4　未知ファミリの検知性能

未知ファミリ	Rig	Neutrino	Magnitude	Sundown
Baseline	0.9089	**0.9894**	0.4655	0.4905
Equal conf.	0.8890	0.9892	0.3757	0.4590
Proposed	**0.9183**	0.9890	**0.5833**	**0.5580**
Oracle	0.9777	0.9897	0.9231	0.8935

表 5　処理時間．括弧内の数値は標準偏差を示す．

	学習 (sec.)	テスト (ms/data)
Baseline	477 (± 14)	0.772 (± 0.057)
Equal conf.	654 (± 15)	0.734 (± 0.046)
Proposed	652 (± 16)	0.759 (± 0.015)

法よりも高い検知性能を実現した．

各手法の検知性能をより詳細に比較するため，ROC 曲線を図 6 に示す．図 6 から提案手法の検知率が baseline と比較してどの程度向上するかは，未知と想定するファミリだけでなく誤検知率によっても異なっていることが分かる．提案手法によって最も検知率が向上したのは，Magnitude を未知と想定した誤検知率 3.2%のときであり，検知率は baseline と比較して 19%向上していた．ファミリの類似度を考慮しない equal confusion を適用した場合は，baseline よりも検知性能が低下していた．これは，特徴が大きく異なるファミリに共通の特徴を分類に用いたため，未知ファミリの検知に有効な特徴が含まれていなかったことが原因だと考えられる．

5.3　処理時間

各手法の学習とテストに必要な処理時間を表 5 に示す．実験は，2 core CPU，8GB RAM の Ubuntu がインストールされた端末で実施した．提案手法と equal confusion は adversarial training を適用しているため，学習に必要な時間が baseline よりも長くなっていた．テストでは，baseline と処理は同じであるため，分類に必要な時間は類似していた．評価に用いたテストデータには，3 ヶ月間以上のデータが含まれているため，提案手法では最低でも 3 ヵ月間は高い検知性能を達成することができる．提案手法の学習時間は baseline と比較して増加しているが，検知性能を維持できる時間よりも非常に短いため，提案手法を実システムに適用した場合でも，高い検知性能を達成できる．

6.　考察

提案手法の適用可能性　本稿では，提案手法を悪性通信系列検知システムに適用して評価を実施した．しかし，提案手法によって未知ファミリの検知性能向上が期待できる条件は，悪性データが攻撃ツールで作成されていることである．サイバーセキュリティの多くの悪性データはこの条件を満たすため，アンドロイドアプリ [1] や悪性サイト [3] 等を対象とした悪性検知システムに適用することで，これらのシステムの未知ファミリの検知性能を向上させることができると考えられる．

提案手法での未知ファミリの検知性能　提案手法によって未知ファミリの検知性能がどの程度向上するかは，教師データに含まれるファミリ同士の類似度と，未知ファミリと既知ファミリの類似度に依存している．大きく特徴が異なるファミリしか含まれないデータセットと，既知ファミリと共通の特徴が存在しない未知ファミリに対しては，提案手法で検知性能を向上させることができない．しかし，5 章の評価では，互いに類似するファミリが多いため，未知と想定するファミリによらず，多くの場合で検知性能を向上させられることを示した．

評価の妥当性 本稿の評価は 1 つのファミリを未知と想定して検知性能を評価している．しかし，本来想定される未知ファミリの検知の状況は，攻撃者によって新たに作成されたファミリの検知である．5 章で示したように，最初に検知された時期が異なるファミリであっても共通の特徴を持つため，新たなファミリも同様に共通の特徴を持つ可能性が高いと考えられる．このため，実際の未知ファミリの検知でも同様の効果が期待できる．

ファミリラベルの収集 提案手法では，悪性/良性のラベルの他にファミリのラベルも必要である．ファミリラベルは IDS や Anti-virus の検知名に基づいて収集することが可能である [11, 12]．しかし，これらのラベルは検知されたばかりの悪性データに関しては，正確なラベルが得られない場合があることが知られている [13]．5 章の評価では，提案手法は 3 ヶ月間以上のテストデータに対して高い検知性能を実現している．言い換えると，教師データに 3 ヵ月以上前のデータしか含まれていなくても高い検知性能を実現可能である．このため，正確なファミリラベルが得られるデータのみを教師データとして使用すれば十分である．

7. 関連研究

機械学習を応用した悪性検知手法 機械学習を応用した悪性検知手法は，アンドロイドアプリ [1]，IoT 機器のファームウェア [2]，悪性サイト [3] 等の様々なデータに対し提案されている．しかし，これらの研究は既知の悪性データに類似のデータを検知することを目的としており，未知ファミリの検知性能向上については検討していない．

悪性データの特徴が変化することを想定し，変化に影響されにくい特徴量の設計手法 [14] も提案されている．この手法では，特徴量がどのように変化するかが仮定されているが，未知ファミリの検知では，特徴量の変化を仮定することができない．そのため，未知ファミリにも含まれている可能性が高い特徴量に基づく分類器を構築するために，類似のファミリに共通の特徴に基づくて手法を提案した．

Adversarial Training Adversarial training は，データセットに共通の representation の学習手法として，domain adaptation に応用されている [6,7]．これらの手法では，分類対象のラベルなしデータが入手できることが想定されているが，本稿の分類対象である未知ファミリのデータを事前に入手することはできない．さらに，domain adaptation では 2 つのデータセットの分類が同等に失敗するように最適化を実施するが，提案手法では，類似しているファミリの分類が失敗するように最適化を実施している．

8. おわりに

教師データに含まれない新たな攻撃ツールで作成された悪性データ（未知ファミリ）の検知性能を向上させるために，本稿では，類似する既知ファミリに共通の特徴に基づく分類器を構築する手法を提案した．分類器の学習には，domain adaptation で用いられている adversarial training を応用した．具体的には，ニューラルネットワークを用いて悪性/良性の分類を学習するのと同時に，ニューラルネットワークの中間層の値（representation）に基づいて既知ファミリを分類した際に，類似のファミリの分類が失敗するように representation を更新する．提案手法を悪性通信系列検知システムに適用した評価では，一般的なニューラルネットワークと比較して，教師データに含まれないファミリの検知率を最大 19%向上させられることを示した．

参考文献

[1] Mariconti, E. et al.: Mamadroid: Detecting Android Malware by Building Markov Chains of Behavioral Models, *Proc. of the 2016 Network and Distributed Syst. Security Symp.* (2016).

[2] Xu, X. et al.: Neural Network-based Graph Embedding for Cross-platform Binary Code Similarity Detection, *Proc. of the 24th ACM Conf. on Comput. and Commun. Security*, pp. 363–376 (2017).

[3] Taylor, T. et al.: Detecting Malicious Exploit Kits using Tree-based Similarity Searches, *Proc. of the 6th ACM Conf. on Data and Application Security and Privacy*, pp. 255–266 (2016).

[4] Symantec: Internet Security Threat Report, https://www.symantec.com/security-center/threat-report.

[5] Canali, D. et al.: Prophiler: a Fast Filter for the Large-scale Detection of Malicious Web Pages, *Proc. of the 20th Int. Conf. on World Wide Web*, pp. 197–206 (2011).

[6] Ajakan, H. et al.: Domain-adversarial Neural Networks, *arXiv preprint arXiv:1412.4446* (2014).

[7] Bousmalis, K. et al.: Domain Separation Networks, *Proc. of the 29th Advances in Neural Inform. Process. Syst.*, pp. 343–351 (2016).

[8] Shibahara, T. et al.: Malicious URL Sequence Detection using Event De-noising Convolutional Neural Network, *Proc. of the 2017 IEEE Int. Conf. on Commun.*, pp. 1–7 (2017).

[9] Srivastava, N. et al.: Dropout: a Simple Way to Prevent Neural Networks from Overfitting, *J. of Machine Learning Res.*, Vol. 15, No. 1, pp. 1929–1958 (2014).

[10] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *arXiv preprint arXiv:1412.6980* (2014).

[11] Open Information Security Foundation: Suricata, http://www.broadanalysis.com/.

[12] Sebastián, M. et al.: AVClass: A Tool for Massive Malware Labeling, *Int. Symp. on Res. in Attacks, Intrusions, and Defenses*, pp. 230–253 (2016).

[13] Kantchelian, A. et al.: Better Malware Ground Truth: Techniques for Weighting Anti-virus Vendor Labels, *Proc. of the 8th ACM Workshop on Artificial Intelligence and Security*, pp. 45–56 (2015).

[14] Bartos, K. et al.: Optimized Invariant Representation of Network Traffic for Detecting Unseen Malware Variants., *Proc. of the 25th USENIX Security Symp.*, pp. 807–822 (2016).