畳み込みニューラルネットワークを用いたURL系列に基づくドライブバイダウンロード攻撃検知

山西 宏平[1,a)]　芝原 俊樹[2]　高田 雄太[2]　千葉 大紀[2]　秋山 満昭[2]　八木 毅[2]　大下 裕一[1]
村田 正幸[1]

概要：本稿では，プロキシログに含まれる宛先 URL の系列から，ドライブバイダウンロード攻撃に関する悪性 URL 系列を検知する手法を，系列データの解析に優れる Convolutinal Neural Network（CNN）を拡張することで提案する．プロキシログを解析する際，ログには必ず良性サイトの URL が混在するため，文章などの系列データと同様に CNN を適用すると，良性 URL も含んだ系列の特徴を学習する可能性が高い．そこで，提案する Event De-noising CNN (EDCNN) では，良性 URL による悪影響を軽減するために，一定範囲内に出現する 2 つの URL を畳み込むことで悪性な URL 同士を確実に畳み込む．インターネット上の良性サイトや悪性サイトへアクセスした結果を用いた評価では，EDCNN は従来手法と比較して検知性能が高く，さらに過学習の問題を解決することで検知性能を向上できることが明らかになった．

キーワード：ドライブバイダウンロード攻撃，系列データ分類, 畳み込みニューラルネットワーク

1. はじめに

多くの研究者やベンダが対策に取り組んでいるにもかかわらず，ドライブバイダウンロード攻撃によるマルウェア感染は後を絶たない．攻撃者は人気ウェブサイトを改ざんすることで，アクセスしたユーザを複数の URL を経由して，ブラウザやプラグインの脆弱性を悪用する攻撃を実施する攻撃 URL へと転送する [1]．このリダイレクションチェインと呼ばれる構造を用いることで，攻撃者は転送や攻撃といった機能を分離できる．この結果，攻撃者は転送や攻撃の機能を担う URL を短時間に変更し，研究者やベンダによる解析や対策を困難にさせている [1]．さらに，転送過程の URL においてブラウザのフィンガープリンティングによりユーザのクライアント環境を識別し，環境に応じて転送先を攻撃 URL や一般の良性ウェブサイトへと制御することもできる [2]．このような環境依存攻撃は，研究者やベンダによる悪性コンテンツの収集を妨害して対策を困難にさせる効果を持つ．

ドライブバイダウンロード攻撃に起因する被害の抑制には，リダイレクションチェインに関わる一連の URL への転送を防止する感染前の対策 [3] と，個々のネットワークの管理者が悪性ウェブサイトへアクセスしたユーザを通信ログから特定して攻撃の影響を最小限に抑制する感染後の対策がある．前者では，一般的に悪性ウェブサイトに関連する URL やドメインを掲載したブラックリストによる対策が広く実施されている．このブラックリストを作成するためには，意図的に攻撃を受けて悪性ウェブサイトを検知するための，おとりのブラウザであるハニークライアント [4] が活用されている．しかし，前述のように，攻撃に利用される URL の変動や環境依存攻撃の汎用化に伴い，ユーザが悪性ウェブサイトに通信する以前にハニークライアントが悪性ウェブサイトを検知することが困難となっている．このため，感染後の対策が必要とされている．

通信ログから悪性ウェブサイトへのアクセスを検知する手法として，ユーザが取得したウェブコンテンツを解析する手法が検討されている（例えば [5]）．しかし，これらの手法では，ユーザが取得したウェブコンテンツを網羅的に収集する必要がある．ネットワーク管理者にとって，網羅的にウェブコンテンツを収集して解析することは，人やコンピュータ等の様々なリソースを必要とすることから非現実的である．一方，プロキシログのようなアクセス履歴情報の記録・解析は一般的に行われており，このログからは，ウェブコンテンツの情報は得られないものの，ユーザのアクセス先の URL の系列（URL 系列）を得ることは可能である．ドライブバイダウンロード攻撃が行われている場合の URL 系列にはリダイレクションチェインを構成す

[1] 大阪大学大学院情報科学研究科
[2] NTT セキュアプラットフォーム研究所
[a)] k-yamanishi@ist.osaka-u.ac.jp

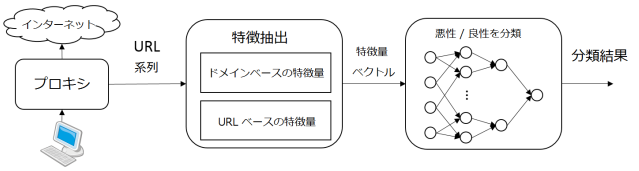

図 1　提案手法の処理の流れ

る複数の URL が含まれており，それらの URL が持つ特徴や順序関係には規則性が存在する [6]．そこで，本稿では，URL 系列からのドライブバイダウンロード攻撃検知を目標とする．

URL 系列中のリダイレクト関係にある URL 同士の関係は，文章中の単語同士の構文的依存関係と類似している．文章の解析に代表される自然言語処理分野では，畳み込みニューラルネットワーク (CNN) が高い精度を実現している [7]．CNN は，文章中の単語系列における局所的な特徴を活用し，徐々により広い範囲の特徴抽出を繰り返すことで，文章全体の特徴を捉え，意味推定や予測変換を実現する．このため，自然言語処理分野と同様に，URL 系列がドライブバイダウンロード攻撃を含むもの (悪性) であるか攻撃を含まないもの (良性) であるかの分類にも CNN による構造解析が効果的である可能性が高い．しかし，文章とは異なり，リダイレクションチェインを含む URL 系列（悪性 URL 系列）には，ドライブバイダウンロード攻撃に関連した悪性ウェブサイトの URL（悪性 URL）だけでなく，改ざんされた人気ウェブサイトが読み込むウェブサイトの URL（良性 URL）も含まれている．このような良性 URL は悪性 URL 系列を判定する上で悪影響を及ぼす可能性がある．そこで我々は，良性 URL の影響を削減するよう CNN のネットワーク構造を改良した Event De-noising CNN(EDCNN) を提案する．EDCNN は，URL 系列の部分系列ではなく，近接する 2 つの URL から特徴を抽出するネットワークを構築する．これにより，リダイレクト関係にある URL 同士を畳み込める可能性を高めることができるため，URL 系列に含まれる良性 URL の検知精度に対する影響を削減し，悪性 URL 系列を正確に検知できる．

EDCNN の有効性を検証するために，1 年間にわたり収集した悪性 URL 系列と良性 URL 系列を用いて，代表的な悪性判定手法によく用いられるランダムフォレスト（RF）や CNN と EDCNN を比較した．評価の結果，EDCNN は RF や CNN と比較して検知性能が高いことが明らかとなった．さらに，CNN や EDCNN に過学習の問題があり，この問題を解決することで検知率を向上できる可能性があることが明らかとなった．

2. 提案手法

2.1 提案手法を適用するシステム構成

提案手法は，URL 系列を入力として判定結果を出力する，図 1 に示すシステムに実装されることを想定している．URL 系列は，プロキシログに記録されたアクセス先 URL の部分列であり，判定結果は，悪性 (系列中に悪性なリダイレクションチェインが存在する) または良性である．

宛先 URL の評価では，宛先ドメインの階層構造や対応する IP アドレスの時系列などに着目して特徴量を抽出して，機械学習を適用することで悪性判定が実施されてきた [3,8]．このため，提案手法でも，従来から用いられていた特徴量を抽出し，特徴量の系列を CNN で識別する．次節以降で，特徴量の抽出と CNN での識別について述べる．

2.2 URL の特徴抽出

URL の悪性判定に有効な特徴量としては，ドメインベースの特徴量と URL ベースの特徴量が提案されている [3,8]．提案手法では，両者の観点から特徴量を設計することで，検知性能の向上を目指す．本稿で用いた特徴量を下記に示す．ただし，提案手法で用いる特徴量は下記のものに限定されないため，他の特徴量を加えることも可能である．

2.2.1 ドメインベースの特徴量

ドメインベースの特徴量としては，ドメインと IP アドレスの対応関係に着目した特徴量 [3] や，ユーザの DNS 解決の挙動に着目した特徴量 [9] が提案されている．しかし，後者は，一般的に入手が困難である，権威 DNS サーバのログ等が必要となる．このため，実現容易性の観点から提案手法では前者を適用する．

具体的には，まず各入力 URL のドメイン部分を抽出し，表 1 に示すドメインベースの特徴量を抽出する．Notos [3] の場合と同様に，ドメインベースの特徴量は，過去に解決された IP アドレス（RHIP: Related Historic IP addresses）とドメイン名（RHDN: Related Historic Domain Names）から抽出される情報で構成される．RHIP は，完全修飾ドメイン名（FQDN），サードレベルドメイン（3LD），セカンドレベルドメイン（2LD）のそれぞれに対して過去に解決されたすべての IP アドレスの集合を意味する．ドメインごとに RHIP を取得し，表 11-18 番の，18 個の RHIP の特徴量を抽出する．一方，RHDN は，対象のドメイン名に紐付いた IP アドレスと同一の AS 内の IP アドレスを過去に利用したことのあるドメイン名の集合を抽出することで得ることができる．これは，対象のドメイン名のネットワーク的近傍に存在するドメイン名を抽出する意味を持つ．ドメインごとに RHDN を取得し，表 1 の 19-35 番の，17 個の RHDN の特徴量を抽出する．

2.2.2 URL ベースの特徴量

URL ベースの特徴量としては，様々な特徴量が提案されている (例えば [8]) 本稿では，これらの特徴量の中から，多くの手法において用いられていた，表 2 に示す特徴量を選定して使用した．

表 1　ドメインベースの特徴量

ID	RHIP の特徴量	ID	RHDN の特徴量
1	BGP prefix 数 (FQDN)	19	FQDN 数
2	BGP prefix 数 (3LD)	20	長さの平均
3	BGP prefix 数 (2LD)	21	長さの標準偏差
4	登録国の総数 (FQDN)	22	出現頻度の平均 (1-gram)
5	登録国の総数 (3LD)	23	出現頻度の中央値 (1-gram)
6	登録国の総数 (2LD)	24	標準偏差 (1-gram)
7	IP アドレス数 (3LD)	25	出現頻度の平均 (2-grams)
8	IP アドレス数 (2LD)	26	出現頻度の中央値 (2-grams)
9	組織数 (FQDN)	27	出現頻度の標準偏差 (2-grams)
10	AS 番号の総数 (FQDN)	28	出現頻度の平均 (3-grams)
11	AS 番号の総数 (3LD)	29	出現頻度の中央値 (3-grams)
12	AS 番号の総数 (2LD)	30	出現頻度の標準偏差 (3-grams)
13	レジストリ数 (FQDN)	31	TLD 数
14	レジストリ数 (3LD)	32	.com の割合
15	レジストリ数 (2LD)	33	出現頻度の平均 (TLD)
16	登録後の日数 (FQDN)	34	出現頻度の中央値 (TLD)
17	登録後の日数 (3LD)	35	出現頻度の標準偏差 (TLD)
18	登録後の日数 (2LD)		

表 2　URL ベースの特徴量

ID	特徴量
1	ドメインの長さ
2	ファイル名の長さ
3	URL の長さ
4	パスの長さ
5	公開ブラックリスト掲載の有無
6	ファイル名内に知られた悪性パターンの登場の有無
7	サブドメインの有無
8	URL 中の IP アドレスの有無
9	URL 中のポート番号の有無
10	ドメインに対応する IP アドレスの総数
11	AS 番号
12	IP アドレスに対応した経度
13	TLD
14	IP アドレスに対応した国名
15	IP アドレスに対応した都市名

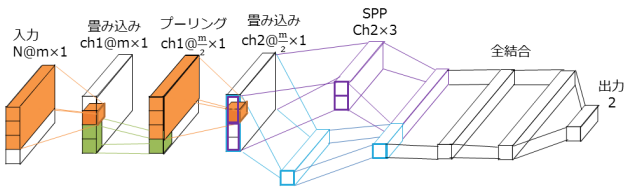

図 2　CNN の構造．*ch*1 と *ch*2 は畳み込み層のチャネル数を指す．

2.3　CNN を用いた識別

2.3.1　Convolutional Neural Network

本稿で提案する EDCNN のベースとなる CNN の全体構成を図 2 に示す．リダイレクトチェインには 3 個から 4 個の悪性 URL が含まれていることが多いため，これらの悪性 URL の情報を統合するために，本稿では，畳み込みとプーリングを 2 回実施している．その後，2 つの全結合層を重ね，抽出された情報を統合する．このネットワークは，6 種類の層から構成され，全部で 8 層構造となっている．以下各層について説明する．

2.3.1.1　入力層

URL から抽出された特徴量を CNN の入力として整形するための層である．特徴量に順序関係は存在しないため，特徴ベクトルを順序関係のないデータとして扱う．また，URL 同士は時系列順に並べる．このため，各 URL を行数 1，列数 1，チャネル数 N のデータとして表現する．N は特徴ベクトルの次元数である．以後，利便性のために，行数 i，列数 j，チャネル数 k の 3 次元データを $k@i \times j$ と書く．TLD や国名のようにカテゴリで表現されている特徴量はワン・ホット表現を用いてベクトル化する．また，特徴量は平均が 0，分散が 1 となるように正規化する．URL 数を m とすると，入力は $N@m \times 1$ のデータとして表される．

2.3.1.2　畳み込み層

畳み込み層の各ニューロンでは，前層の局所的な領域から出力を算出する．このとき，c 番目のチャネルの出力を算出する畳み込みにおいて，基準となる行と列から h 行 0 列離れた k 番目のチャネルに対する重みを $w^c_{h,0,k}$ とする．前層の i 行 1 列 k チャネルの値を $x_{i,1,k}$ とすると，i 行 1 列 c チャネルの出力 $y_{i,1,c}$ は，

$$y_{i,1,c} = \tanh\left(\sum_{h=-H}^{H}\sum_{k=1}^{ch_{pre}} w^c_{h,0,k} x_{i+h,1,k}\right) \qquad (1)$$

で表現される．ここで，H は前層における畳み込みで考慮する中心の行からの距離，ch_{pre} は前層のチャネル数である．このとき，x が前層の領域外の場合は 0 として扱う．ここで，ch_{conv} を畳み込み層の出力チャネル数とすると，入力が $ch_{pre}@m \times 1$ であった場合，出力は $ch_{conv}@m \times 1$ となる．この層では，局所的な関係性に基づく処理が可能である．

2.3.1.3　プーリング層

この層では，前層の一部の領域の最大値を出力する．前層で中心となる行から H 離れた行まで考慮すると，出力 $y_{i,1,k}$ は，

$$y_{i,1,k} = \max_{-H \le h \le H}(x_{i+h,1,k}) \qquad (2)$$

で表される．このとき，x が前層の領域外の場合は 0 として扱う．入力が $ch@m \times 1$ であった場合，出力は $ch@\frac{m}{2} \times 1$ となる．これにより，入力の位置が少しずれていても出力が変わらなくなる．

2.3.1.4　空間ピラミッドプーリング層

この層は，任意の大きさの入力に対し，固定の長さのベクトルを出力する [10]．まず，前層を格子状に 4 分割，16 分割と分割していく．その後，分割された領域から最大値を選択する．最後に，各領域の最大値を並べたベクトルを出力する．列数が 1 の場合は，行方向にのみ分割する．4 分割まで考慮する場合，入力が $ch@m \times j, (j \ge 2)$ であった場

合，出力は大きさ $5ch$ のベクトルとなる．入力が $ch@m\times1$ であった場合，出力は大きさ $3ch$ のベクトルとなる．この層では，位置関係の情報を保持しつつ重要な情報を上位の層に伝えることができる．

2.3.1.5 全結合層

この層では，前層全体から出力を算出する．前層のニューロンとこの層のニューロンは，互いに他の層のすべてのニューロンと結合している．この層の出力は，前層の i 番目のニューロンの値を x_i，この層の j 番目のニューロンの値を y_j とし，この間の重みを w_{ij} とすると，出力は

$$y_j = \tanh\left(\sum_i w_{ij}x_i\right) \tag{3}$$

で表される．学習時には一部のニューロンをランダムに無視することで汎化性能を向上させるドロップアウト [11] を適用する．この層では，前層に伝播されたすべての情報を統合して，出力を算出することができる．

2.3.1.6 出力層

この層では，最終的な識別結果を出力する．出力層のニューロンの数は，識別するクラス数であり，ネットワークの構成は，全結合層と同じく完全二部グラフである．j 番目の出力層のニューロンと i 番目の前層のニューロンの重みを w_{ij} とし，出力 z_j を $z_j = \sum_i w_{ij}x_i.$ で定義する．j 番目の出力 y_j は，ソフトマックス関数を用いて下記のように算出される．

$$y_j = \frac{e^{z_j}}{\sum_i e^{z_i}} \tag{4}$$

このように出力を算出することで，出力層のすべてニューロンで値が正，総和が 1 となるため，識別対象のデータが各クラスに属する確率として扱うことができる．

CNN の各層の重みの学習は，誤差逆伝播法によって出力層の誤差を逆に伝播させて実施する．最適化には，学習率を自動調整することで学習時間の短縮が可能な AdaDelta [12] を用いる．

2.3.2 Event De-noising Convolutional Neural Network

本稿では，悪性 URL 系列に含まれる良性 URL の悪影響を削減するよう CNN を拡張した，提案の Event De-noising CNN (EDCNN) を適用する．EDCNN では，ドライブバイダウンロード攻撃に関係する悪性な URL 同士の畳み込みが実施できるよう，URL 系列の部分系列を畳み込むのではなく，系列の近くに出現する 2 つの URL からの畳み込みを実施する．本畳み込みを実現するために，CNN の畳み込み層の入力を並び替えるアロケーション層を導入する．

EDCNN の全体構成を図 3 に示す．このネットワークは，7 種類の層から構成され，全部で 10 層構造となっている．以下 EDCNN で新しく導入したアロケーション層と，変更を加えた畳み込み層，プーリング層について説明する．

2.3.2.1 アロケーション層

悪性な URL は，URL 系列において，隣接しているか，良性 URL が混在しつつも近接している可能性が高い．そこで，図 4 に示すように，系列中の i 番目の URL に対し，R を畳み込みを行う URL 間隔の上限として，i 番目の URL と $i+1$ 番目の URL，i 番目の URL と $i+2$ 番目の URL...i 番目の URL と $i+R$ 番目の URL が隣接するように URL の配置を行う．これを各行で実施する．入力を x とすると，出力 y は，

$$y_{2i,j,k} = \begin{cases} x_{i,1,k} & (i+j \le m) \\ 0 & (otherwise) \end{cases} \tag{5}$$

$$y_{2i+1,j,k} = \begin{cases} x_{i+j,1,k} & (i+j \le m) \\ 0 & (otherwise) \end{cases} \tag{6}$$

と表すことができる．ここで，m は入力の行数である．入力が $N@m\times1$ であった場合，出力は $N@2(m-1)\times R$ となる．

2.3.2.2 畳み込み層

アロケーション層では，$2i$ 行 j 列のデータと $2i+1$ 行 j 列のデータを畳み込むように並び替えを実施した．このため，畳み込み層では，これらのデータの畳み込みを実施する．c 番目のチャネルの出力を算出する畳み込みにおいて，基準となる行と列から h 行 0 列離れた k 番目のチャネルに対する重みを $w^c_{h,0,k}$ とする．前層の i 行 j 列 k チャネルの値を $x_{i,j,k}$ とすると，i 行，j 列，c チャネルの出力 $y_{i,j,c}$ は，

$$y_{i,j,c} = \tanh\left(\sum_{h=0}^{1}\sum_{k=1}^{N} w^c_{h,0,k}x_{2i+h,j,k}\right) \tag{7}$$

畳み込みのウィンドウを行方向には 2 ずつ移動させるため，入力が $N@2(m-1)\times R$ であった場合，出力は $N@(m-1)\times R$ となる．

2.3.2.3 プーリング層

列方向は各リダイレクト候補の畳み込み結果であるが，この中に含まれる悪性 URL 間のリダイレクトは 1 つ以下である．このため，各列において最も悪性 URL 間のリダイレクトらしい値を選択することで悪性 URL 系列の判定に効果があると考えられる．そこで，プーリングを行う際にすべての列が含まれるように実施する．前層で中心となる行から H 離れた行まで考慮すると，出力 $y_{i,1,k}$ は，

$$y_{i,1,k} = \max_{0\le j\le R}(x_{i+h,j,k}). \tag{8}$$

で表される．このとき，x が前層の領域外の場合は 0 として扱う．入力が $N@(m-1)\times R$ であった場合，出力は $N@(m-1)\times1$ となる．

重みの学習は CNN と同様に誤差逆伝播法で実施する．アロケーション層における誤差逆伝搬は，他の層と同様に結合しているニューロンに誤差を伝播させることで実現できる．

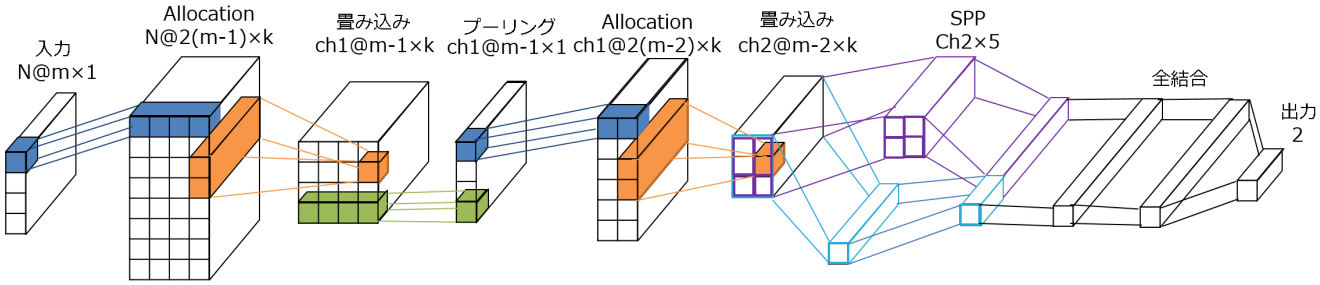

図 3　EDCNN の構造．*ch*1 と *ch*2 は 畳み込み層のチャネル数を示す．

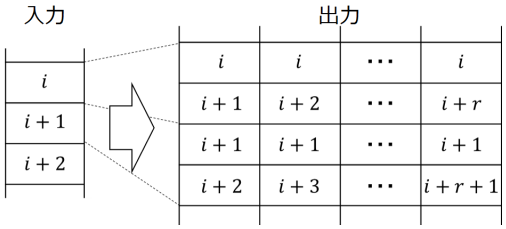

図 4　アロケーション層

表 3　データセットの概要

	教師データ			テストデータ		
	期間	悪性	良性	期間	悪性	良性
1	2015/2～7	3,039	5,905	2015/8	399	967
2	2015/3～8	3,182	5,905	2015/9	352	957
3	2015/4～9	3,126	5,892	2015/10	158	956
4	2015/5～10	2,785	5,895	2015/11	123	958
5	2015/6～11	2,535	5,880	2015/12	153	970
6	2015/7～12	1,906	5,891	2016/1	119	951

3. 評価

ハニークライアントが検知する以前の悪性ウェブサイトへのアクセスが含まれる URL 系列を検知可能か評価を実施した．このために，ある時点以前に検知された悪性ウェブサイトへのアクセスが含まれる URL 系列を学習データとして使用し，それ以降に検知された悪性ウェブサイトへのアクセスが含まれる URL 系列を識別した際の性能を評価した．

プロキシログから取得された URL 系列には，複数のウェブサイトへの通信が含まれていることや，ウェブサイトの一部分しか含まれていないことが考えられる．このように識別に悪影響を与える要素が多数存在している状況下での評価では，提案手法の有効性や制約を解明することが困難である．そこで，本評価では提案手法の最初の評価として，一つのウェブサイトへのアクセスのみが含まれる URL 系列が入力として与えられる状況下で評価した．プロキシログに提案手法を適用する際には，識別に必要な計算時間も重要となるため，計算時間の評価も実施した．さらに，識別性能を改善可能か調査するために，識別器の学習において過学習が発生しているかと，その原因について解析した．

本評価では，URL 系列の識別にランダムフォレスト（RF），CNN，EDCNN を適用した際の性能を比較した．RF は長さの異なる系列データを入力できないため，RF による性能評価では系列中の個々の URL を識別し，その結果を統合して URL 系列の識別結果を出力した．系列中の 1 つ以上の URL が悪性と識別された場合に，その URL 系列を悪性と判定した．RF の悪性教師データとして攻撃 URL を用いることが適切であるが，JavaScript，PDF，Flash 等の多くの種類のコンテンツがエクスプロイトに使用されているため，エクスプロイト URL を網羅的に検出することは困難である．このため，悪性 URL 系列中の URL すべてを悪性とラベルづけを行い RF の学習に用いた．

3.1　データセット

一つのウェブサイトに関する URL 系列を用いた評価を実施するために，ハニークライアント [4] でウェブページを解析した際の URL の系列を収集した．公開ブラックリスト [13,14] と alexa [15] に記載されているウェブサイトを対象とし，2015 年 8 月～2016 年 1 月に収集を実施した．本評価では，1 ヶ月ごとに再学習を実施し，教師データはテストデータの直近 6 ヶ月間に収集された URL 系列を用いた．このとき，教師データとテストデータに同一な URL 系列が含まれている場合は，テストデータから削除した．評価に使用した URL 系列数とデータを収集した期間を表 3 に示す．URL 系列数は合計で，悪性が 17877，良性が 41127 となった．

3.2　ハイパーパラメータの最適化

識別器を構築するためには，ハイパーパラメータを決める必要がある．このために，事前にクロスバリデーションを実施して最も識別性能が高かったパラメータを採用した．CNN と EDCNN で決める必要があるハイパーパラメータを表 4 に示す．クロスバリデーションの結果，CNN と EDCNN のハイパーパラメータはともに，1 段目の畳み込み層のチャネル数は 200，2 段目の畳み込み層のチャネル数は 400，全結合層のニューロンは 1000，畳み込み対象の間隔は 5 となった．さらに，過学習を避けるために学習の早期打ち切りを行うが，学習を実施する最適な回数はデー

表 4　ハイパーパラメータのリスト

パラメータ	値の候補
1 段目の畳み込み層のチャネル数	100, 200, 400
2 段目の畳み込み層のチャネル数	100, 200, 400
畳み込み対象とする URL の間隔の上限	5, 10, 15
全結合層のニューロン数	250, 500, 1000

表 5　検知性能 (値は平均 ± 標準偏差)

Classifier	Precision	Recall	F-measure
RF	0.55 ± 0.06	**0.97 ± 0.04**	0.71 ± 0.04
CNN	0.44 ± 0.10	0.97 ± 0.02	0.59 ± 0.08
EDCNN	**0.64 ± 0.05**	0.90 ± 0.03	**0.72 ± 0.06**

タによってばらつきがあるため，教師データを用いて決定した．具体的には，テストデータに近い時期の URL 系列 1 割を評価に用い，残りのデータを学習に用いた．このときに最も識別性能が高かった学習回数を採用した．

RF のハイパーパラメータとしては，各決定木の学習で用いる特徴ベクトルの次元数，決定木の数を決定する必要がある．これらのハイパーパラメータは，CNN および EDCNN の学習回数と同じ方法で決定した．

3.3　評価結果

3.3.1　識別性能

評価に用いたデータセットには，悪性 URL 系列より良性 URL 系列が多く含まれる．このため，評価指標として F 値，Precision，Recall を用いた．Precision は誤検知の少なさを示す指標であり，Recall は見逃しの少なさを示す指標，F 値は誤検知と見逃しの両方を考慮した指標である．悪性 URL 系列を正しく悪性と判定したデータ数を TP，誤って良性と判定したデータ数を FN，良性 URL 系列を正しく良性と判定したデータ数を TN，誤って悪性と判定したデータ数を FP とすると各指標は下記のように定義される．

$$\text{Precision} = \frac{TP}{TP + FP} \tag{9}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{10}$$

$$\text{F 値} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

テストデータを識別した際の各指標を表 5 に示す．Precision と F 値は，EDCNN が最も高く，Recall は RF が最も高い結果となった．見逃しと誤検知を両方考慮した識別器全体の性能を示す指標である F 値の結果から，URL 系列の識別には EDCNN が最も適した識別器であることが分かった．

さらに，経年劣化に伴う再学習の頻度が 1 ヶ月で十分かを調査するために，1 週間ごとの性能を評価した．悪性 URL 系列では使用されるドメインが頻繁に変更されるため，経年劣化が発生すると考えられるが，良性サイトでは使用されるドメインはほとんど変更されないため，経年劣化は発生しにくい考えられる．このため，1 週間ごとの性能の評価では，悪性サイトを検知できたかを示す指標である Recall を用いた．Recall が向上する期間や低下する期間は存在するが，識別器構築から 1 週後と 4 週後で大きな変化はなかったことから，識別を構築してから 1 ヶ月以内では経年劣化は発生しないと考えられる（図 5）．このため，1 ヶ月ごとに識別器の再学習を実施することで，識別器の性能を維持することが可能だと考えられる．

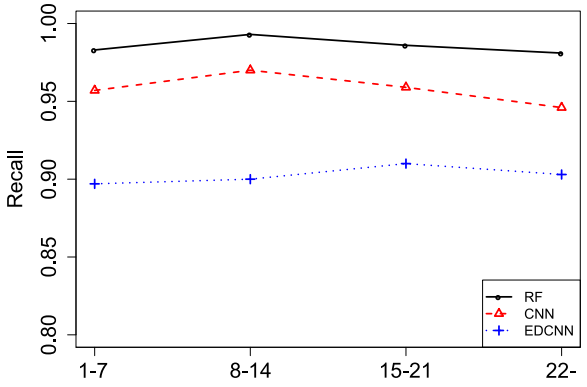

図 5　時間経過による Recall の変化

表 6　計算時間 (秒/URL 系列)

特徴量	抽出時間	識別器	識別時間
ドメイン	0.35	RF	0.21
URL	1.75	CNN	0.09
		EDCNN	0.07

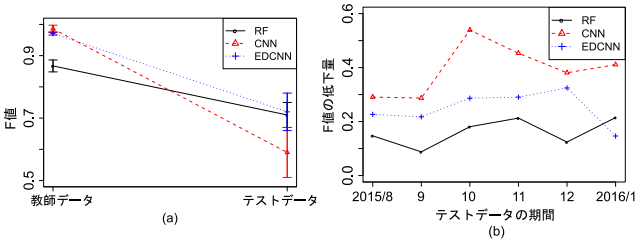

図 6　(a) 教師データとテストデータを識別した際の F 値．(b) 過学習による F 値の減少量の変化．

3.3.2　計算時間

特徴量抽出と識別に必要な時間を計測した．提案手法のプロトタイプは 3 つの異なるマシンを用いて実装されている．URL ベースの特徴量抽出は，12 コア CPUx4，128GB RAM の Ubuntu サーバを用いており，ドメインベースの特徴量抽出は 16 コア CPU，128GB RAM のマスターサーバ 2 台と，16 コア CPU，64GB RAM のスレーブサーバ 16 台を用いる．また，識別処理は，12 コア CPUx4，256GB RAM の Windows PC を用いて行われている．表 6 に本プロトタイプが要した計算時間を示す．表より，本プロトタイプが 1 つの URL 系列の識別を行うのに必要な計算時間は約 2 秒であり，1 日で 43200 サイトの URL 系列を判定可能であることがわかる．

3.3.3　識別器の過学習

機械学習を用いた手法では，識別器の教師データへの過

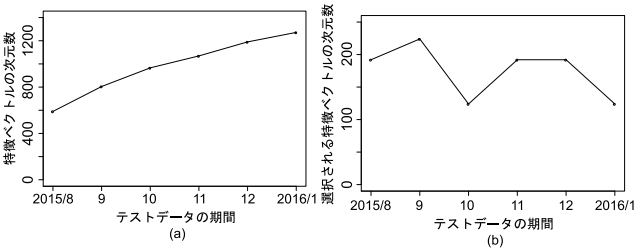

図 7 (a) 特徴ベクトルの次元数．(b) 選択された特徴ベクトルの次元数．

学習が発生することによって，テストデータを識別した際の性能が低下してしまう．このため，本評価において過学習が発生しているか調査した．教師データを識別した際のF値とテストデータを識別した際のF値の平均を図 6(a) に示す．すべての手法において過学習が発生していたが，CNN，EDCNN，RF の順で過学習が大きかった．RF では，指定された数だけランダムに特徴ベクトルの次元を選択して，各決定木を学習させることで本来識別に有効でない特徴ベクトルの次元の影響を小さくし，過学習を抑制している．この結果，過学習が小さかったと考えられる．一方，CNN と EDCNN ではすべての特徴量を使用して学習しているため，過学習が大きかったと考えられる．EDCNN では，CNN に比べ過学習の影響が小さかったが，畳み込みで考慮するニューロンの数が小さいことによって過学習が小さくなったと考えられる．

特徴ベクトルに識別に有効でない次元が含まれており，すべての次元を用いることによって過学習が大きくなっていることを解明するために調査を実施した．具体的には，各データセットにおける特徴ベクトルで識別に有効な次元数と過学習の大きさとの関係を調査した．まず，過学習の大きさを明らかにするために，テストデータを識別した際の F 値の教師データを識別した際の F 値からの減少量を図 6(b) に示す．強い相関ではないが，より新しいテストデータになるにつれて F 値の減少量が増加し，過学習が大きくなる傾向があることが分かった．次に，特徴ベクトルにおける有効な次元の割合を調査した．特徴ベクトルでは，カテゴリの特徴量に関してはワン・ホット表現を使用している．このため，教師データ内で出現するカテゴリ数によって特徴ベクトルの次元数が変化する．そこで，各データセットでの特徴ベクトルの次元数を図 7(a) に示す．この結果から，新しいデータセットほど次元数が大きくなっていることが分かった．さらに，特徴ベクトルの次元数が大きくなった際に，識別に有効となる特徴量ベクトルの次元数が多くなるか調査した．ハイパーパラメータとして事前に調整された RF の各決定木で選択される特徴ベクトルの次元数を図 7(b) に示す．特徴ベクトルの次元数が増加しても，選択される特徴ベクトルの次元数は増加しないことから，新しいデータセットには識別に有効でない特徴ベクトルの次元が多く含まれることが分かった．これらの結果から，特徴ベクトルの中に識別に有効な次元が多いと，過学習が大きくなる傾向があることが分かった．つまり，特徴ベクトル中の識別に有効でない次元の影響を削減することで，CNN と EDCNN の検知性能をさらに向上可能であると考えられる．

4. 制約事項と今後の課題

提案手法は，URL にどのようなコンテンツが対応しているかは考慮せず識別を行う．このため，攻撃者によって攻撃コードが削除された後でも，URL 系列として同一である場合は悪性と判定してしまうと考えられる．提案手法で悪性と判定された URL 系列は，攻撃コードのダウンロードが発生したかを手動の解析で特定する必要がある．ただし，提案手法を用いることにより，手動での解析が必要な URL 系列数を減らすことが可能である．

評価結果から，識別に有効でない特徴ベクトルの次元の影響によって過学習が発生していることが示唆された．このため，特徴ベクトルから識別に有効でない次元を削除することや，有効でない次元の悪影響を削減可能なニューラルネットワークに拡張することによって検知性能を向上可能であると考えられる．今後は，提案手法の検知性能をさらに向上させるために，これらを実現する．

本稿の評価では，一つのウェブサイトへのアクセスのみが含まれる URL 系列を入力として用いている．しかしながら，プロキシログから悪性 URL 系列を特定するためには，ウェブサイトが一部しか含まれていない状況や，複数のウェブサイトへの通信が含まれる状況で識別を実施しなければならない．これらの状況での提案手法の有効性と制約の評価が今後の課題である．

5. 関連研究

5.1 悪性サイト検知手法

悪性サイトを検知するために，ハニークライアントと呼ばれるおとりのシステムが用いられている．ハニークライアントでは，不正なプロセスやファイルシステムへのアクセス [4] などに基づき悪性サイトを検知する．しかし，ハニークライアントの目的は，ウェブサイトを解析し悪性サイトを検知することであり，本研究の目的である通信ログに含まれる悪性サイトへの通信検知とは異なっている．

通信ログに含まれる悪性サイトへの通信検知に適用可能な手法として，ウェブコンテンツやリダイレクトに着目した手法が多く研究されている [1,5]．さらに，攻撃に用いられるコンテンツのリダイレクト元になる URL を特定することで，効率的にハニークライアントで解析する手法も提案されている [16]．これらの手法では，悪性コードの特定や，リダイレクト関係の特定にコンテンツの解析が必要となるが，提案手法はコンテンツの解析を行わずに悪性 URL 系列を検出できる．

悪性ウェブサイトのドメインや URL に着目している研究も進められている．例えば，Antonakakis らは，ドメインと対応する IP アドレスの使われ方に着目した手法 [3] を手案している．これらの手法では，1 つのドメインや URL の識別を行うのに対し，提案手法では，ドライブバイダウンロード攻撃に必須なリダイレクトに着目し，URL 系列の識別を行っている．

5.2 深層学習

近年，深層学習が多くの分野に応用され，高い性能を実現していることで注目を集めている．CNN は畳み込みとプーリングを繰り返し実施するニューラルネットワークであり，自然言語などの系列データにも適用されている [7]．セキュリティの分野では，Borgolte らがウェブページを画像として捉えることで，CNN をウェブサイトの改ざん検知に適用したものの，セキュリティのための CNN の拡張は提案されていない [17]．

本稿では，URL 系列に CNN を適用したが，悪性 URL 系列中の良性 URL は，ノイズのように識別に悪影響を及ぼす．URL 系列の悪性 URL 間に挿入されるノイズは，画像のピクセルに加算されるノイズとは性質が異なるため，画像認識で提案されているノイズ除去を行う手法 [18] は適用できない．このため，本研究では，URL 系列に含まれるノイズの影響を低減する，セキュリティのための新しい CNN を提案した．

再帰型ニューラルネットワーク (RNN) は，再帰的構造を持ったニューラルネットワークであり，セキュリティの分野では，Shin らがバイナリにおける関数の識別適用している [19]．CNN の構造は RNN と比較して，URL 系列の特性に合わせて柔軟に変更することができるため，本稿では悪性 URL 系列の検知に CNN を適用した．

6. おわりに

本稿では，プロキシログに含まれる宛先 URL の系列から，ドライブバイダウンロード攻撃に関する通信が含まれる URL 系列を検知する手法を検討した．URL 系列と文章のデータ構造における類似性に着目し，URL 系列の分類に CNN の適用をした．しかし，悪性 URL 系列にはドライブバイダウンロード攻撃に関係のない良性 URL が多数含まれており，従来の CNN を適用すると良性 URL によって検知性能が低下すると考えられる．そこで，良性 URL の影響を削減するために，系列中の近くに存在する 2 つの URL から畳み込みを実施するように CNN を拡張し EDCNN を提案した．1 年間にわたり収集した URL 系列を用いた評価では，一般的な悪性判定手法であるランダムフォレストや CNN と比較し，EDCNN の検知性能が高いことが示された．さらに，識別器の過学習に着目した解析から，EDCNN は識別に有効でない特徴量の影響を下げることで，性能を改善できることが分かった．

参考文献

[1] Zhang, J. et al.: Arrow: Generating signatures to detect drive-by downloads, *Proc. the 20th international conf, on World wide web*, pp. 187–196 (2011).

[2] Kolbitsch, C. et al.: Rozzle: De-cloaking internet malware, *IEEE Symp. on Security and Privacy*, pp. 443–457 (2012).

[3] Antonakakis, M. et al.: Building a Dynamic Reputation System for DNS., *Proc. the 19th USENIX Security Symp.*, pp. 273–290 (2010).

[4] Akiyama, M. et al.: MARIONETTE: Client honeypot for investigating and understanding Web-based malware infection on implicated websites, *Joint Workshop on Information Security* (2009).

[5] Curtsinger, C. et al.: ZOZZLE: Fast and Precise In-Browser JavaScript Malware Detection., *Proc. the 20th USENIX Security Symp.*, pp. 33–48 (2011).

[6] 重田真義ほか：企業での実環境を考慮したサイバー攻撃検知システムの有効性評価，コンピュータセキュリティシンポジウム 2015 論文集， Vol. 2015, No. 3, pp. 16–23 (2015).

[7] Hu, B. et al.: Convolutional neural network architectures for matching natural language sentences, *Advances in Neural Information Processing Systems*, pp. 2042–2050 (2014).

[8] Canali, D. et al.: Prophiler: a fast filter for the large-scale detection of malicious web pages, *Proc. the 20th international conf, on World wide web*, pp. 197–206 (2011).

[9] Antonakakis, M. et al.: Detecting Malware Domains at the Upper DNS Hierarchy., *Proc. the 20th USENIX Security Symp.*, pp. 1–16 (2011).

[10] He, K. et al.: Spatial pyramid pooling in deep convolutional networks for visual recognition, *Proc. the 13th European conf, on Computer Vision*, pp. 346–361 (2014).

[11] Hinton, G. E. et al.: Improving neural networks by preventing co-adaptation of feature detectors, *arXiv preprint arXiv:1207.0580* (2012).

[12] Zeiler, M. D.: ADADELTA: an adaptive learning rate method, *arXiv preprint arXiv:1212.5701* (2012).

[13] MDL: Malware Domain List, MDL (online), available from ⟨http://www.malwaredomainlist.com/⟩ (accessed 2016-08-09).

[14] MalwareBytes: hpHosts, hosts-file (online), available from ⟨https://hosts-file.net/?s=Download⟩ (accessed 2016-08-09).

[15] Alexa.com: Alexa Top Global Sites, Alexa (online), available from ⟨http://www.alexa.com/topsites/⟩ (accessed 2016-08-09).

[16] Taylor, T. et al.: Cache, Trigger, Impersonate: Enabling Context-Sensitive Honeyclient Analysis On-the-Wire, *Proc. the 23rd Annu. Network and Distributed System Security Symp.* (2016).

[17] Borgolte, K. et al.: Meerkat: Detecting website defacements through image-based object recognition, *Proc. the 24th USENIX Security Symp.*, pp. 595–610 (2015).

[18] Vincent, P. et al.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, *J. of Machine Learning Research*, Vol. 11, pp. 3371–3408 (2010).

[19] Shin, E. C. R. et al.: Recognizing functions in binaries with neural networks, *Proc. the 24th USENIX Security Symp.*, pp. 611–626 (2015).