Vol.2020-DPS-182 No.53
Vol.2020-CSEC-88 No.53
2020/3/13

深層学習モデルを用いたURLに着目したアクセスログ内の悪性Webサイト探索

前橋 祐斗[1]　小澤 誠一[1]　山田 明[2]

概要：近年，Drive-by Download 攻撃やフィッシングを代表とする Web 媒介型攻撃が深刻化しており，このような攻撃を行う悪性サイトの迅速な発見が求められている．NICT 委託研究として実施している WarpDrive プロジェクトでは，Web 媒介型攻撃への対策のため，ユーザが Web サイトを閲覧した際のアクセスログの収集を行っている．大量のアクセスログから悪性サイトを発見するには多大な労力と時間を要するため，迅速かつ効率的に発見する手法が重要となる．本研究では，Character-level CNN を用いて URL から特徴抽出することで，大規模アクセスログから悪性サイトを発見する手法を検討する．2019 年 4 月 30 日の時点で GSB にヒットしなかったアクセスログに対して悪性度を判定した結果，約 2700 件のグレー判定結果が得られ，そのうち 20 件については，のちに GSB で悪性と判定されたことを確認した．

Discovering Malicious Websites from Access Logs of URLs Using Deep Learning Model

Abstract: In recent years, web-based cyberattacks such as Drive-by-Download attacks and phishing is getting more serious. It is soliciting to detect malicious sites as quickly as possible. The WarpDrive project funded by NICT is collecting about 15M access logs of users per day to discover malicious sites. Since the size of access logs is huge, it is very helpful to introduce machine learning for discovering malicious sites efficiently. In this work, we adopt Character-level Convolutional Neural Network (CL-CNN), which extracts features from URLs, to predict the maliciousness of web sites. In our experiment using the access logs on April 30, 2019, we demonstrate that CL-CNN predicted at least 20 malicious sites form about 2700 gray URLs that were not detected by GSB.

1. はじめに

近年，Drive-by Download 攻撃やフィッシングを代表とする Web 媒介型攻撃が深刻化しており，このような攻撃を行う悪性サイトの迅速な発見が求められている．一般的に，ユーザが悪性サイトへアクセスすることを防ぐためにブラックリストが用いられている．しかし，ブラックリストでは既知の悪性サイトしか検知できないという問題点がある．また，悪性サイトは日々増加しているため，ブラックリストの更新には多大な労力が必要である．この問題に対処するために，機械学習を用いた悪性サイト検知に関する研究が盛んに行われている．機械学習によって，既知の悪性サイトと似た特徴を抽出することで，ブラックリストに登録されていない悪性サイトの検知が可能となる．Bilge ら [1] の研究では，ドメインに紐づく IP アドレスの数などに着目し，DNS トラフィックから特徴を抽出することで，高い精度で悪性ドメインを検知できることを示した．しかし，Bilge らの提案手法は Passive DNS データを取得できることが前提の技術である．Saxe ら [2] の研究では，Character-level CNN を用いて，URL 文字列のみから特徴を抽出し，悪性 URL を検知する手法を提案している．一般的に，URL 文字列から特徴を抽出する場合，文字列の長さや文字列に含まれる特殊文字の数といった統計的情報が特徴量として用いられる [3]．これに対し，Character-level CNN を用いることで，あらかじめ人間が特徴量を定義することなく，有効な特徴を学習によって得られる．

本研究では，Saxe らの研究で提案された手法をもとに，URL 文字列に加えて，取得が比較的容易なドメインの有効期間を特徴量として使うことで，悪性サイトをより正確に見つけ出すことを目指す．また，提案手法を用いて，ユー

[1] 神戸大学
[2] KDDI 総合研究所

ザの Web 閲覧履歴から悪性度の高い URL を特定し，その悪性度判定の精度を検証する．

2. Character-level CNN

Character-level CNN は自然言語処理で用いられる CNN の一種である [4]．文字列を機械学習モデルに入力する際，前処理として文字列中の単語を区切る，分かち書きが必要となる．しかし，Character-level CNN は文字列を一文字ずつ区切り，ベクトルに変換するため分かち書きの必要がない．ゆえに Character-level CNN を用いることで，単語同士が空白で区切られていない URL 文字列を効率よく処理することができる．

入力文字列における i 番目の文字は Embedding 層で k 次元ベクトル $\boldsymbol{x}_i$ に変換される．ここで，k はハイパーパラメータである．Embedding 層はモデルの他の部分と同様に学習し，文字とベクトルの対応を決定する．各文字から変換されたベクトルを連結することで，長さ n の文字列は次のように表される．

$$\boldsymbol{x}_{1:n} = \boldsymbol{x}_1 \oplus \boldsymbol{x}_2 \oplus \ldots \oplus \boldsymbol{x}_n \tag{1}$$

ここで，$\oplus$ は連結を意味する演算子である．次に，畳み込み層によって特徴マップを生成する．画像認識の分野で利用される CNN は 2 次元の畳み込みを行うのに対して，Character-level CNN は 1 次元の畳み込みを行う．文字列中のウィンドウ $\boldsymbol{x}_{i:i+h-1}$ とフィルタ $\boldsymbol{w} \in \mathbb{R}^{hk}$ の畳み込みによって得られる特徴 c_i は次式で表される．

$$c_i = f(\boldsymbol{w} \cdot \boldsymbol{x}_{i:i+h-1} + b) \tag{2}$$

ここで，b はバイアス項，f は活性化関数である．文字列中の全てのウィンドウ $\{\boldsymbol{x}_{1:h}, \boldsymbol{x}_{2:h+1}, \ldots, \boldsymbol{x}_{n-h+1:n}\}$ にフィルタを適用することで，次式で表される特徴マップ $\boldsymbol{c}$ を得られる．

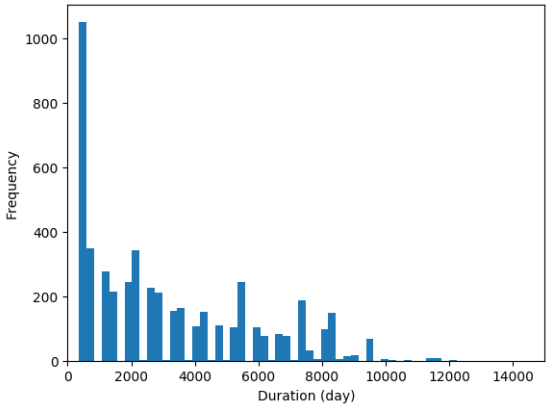
$$\boldsymbol{c} = [c_1, c_2, \ldots, c_{n-h+1}] \tag{3}$$

モデルは異なるウィンドウサイズのフィルタを用いることで，多様な特徴を抽出することができる．続いてプーリング層では，Max Pooling によって特徴マップの要素から最大値のみを選択する．これにより，各特徴マップから最も重要な情報を抽出することができる．各フィルタの出力を連結したものが全結合層への入力となる．

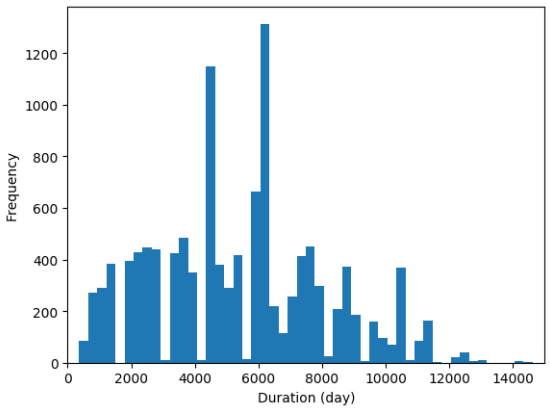
3. 提案手法

3.1 ドメインの有効期間

Saxe らの手法では URL の文字列のみから特徴を抽出していた．本研究では，追加の特徴量としてドメインの有効期間に着目した．本研究で用いたデータセットにおける，悪性 URL と良性 URL に含まれるドメインの有効期間の分布を図 1 に示す．また，表 1 に示すドメインの平均有効期間から，悪性 URL に含まれるドメインは良性 URL に含まれるドメインよりも有効期間が短い傾向にあることがわかる．これは，攻撃者が悪性サイトで使用するドメインを頻繁に変えることで，ブラックリストによる検知を逃れるためであると考えられる．そこで，本実験ではドメインの有効期間を特徴量として追加した悪性 URL 検知モデルを作成し，その性能を評価した．



(a) 悪性 URL



(b) 良性 URL

図 1 ドメインの有効期間: (a) 悪性 URL，(b) 良性 URL

表 1 ドメインの平均有効期間

	件数	平均有効期間（日）
悪性 URL	4905	3140
良性 URL	11834	5365

3.2 悪性 URL 検知システム

本実験で用いたモデルの構造を図 2, 3 に示す．図 2 のモデルでは，一文字ずつ区切られた URL が入力され，Embedding 層でそれぞれベクトルに変換される．畳み込み層とプーリング層では特徴マップを生成し，各フィルタの特徴マップを連結したものが全結合層に入力される．図 3 のモデルでは，図 2 のモデルと同様の処理を行った後，全結合層の途中でドメインの有効期間を入力する．入力の長さは，Saxe ら [2] の研究で用いられた $n = 200$ で固定とした．また，Embedding 層で作成されるベクトルの次元は

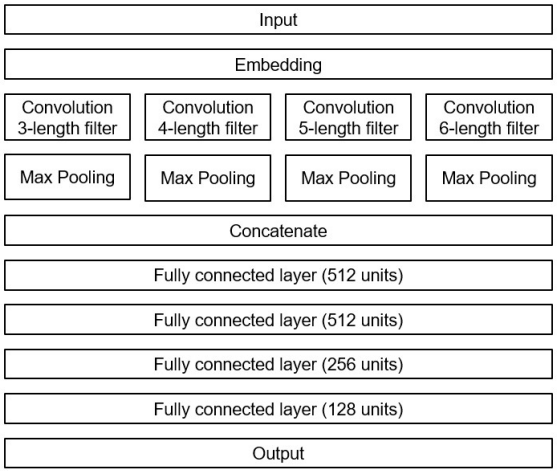


図 2 文字列のみを特徴量としたモデル

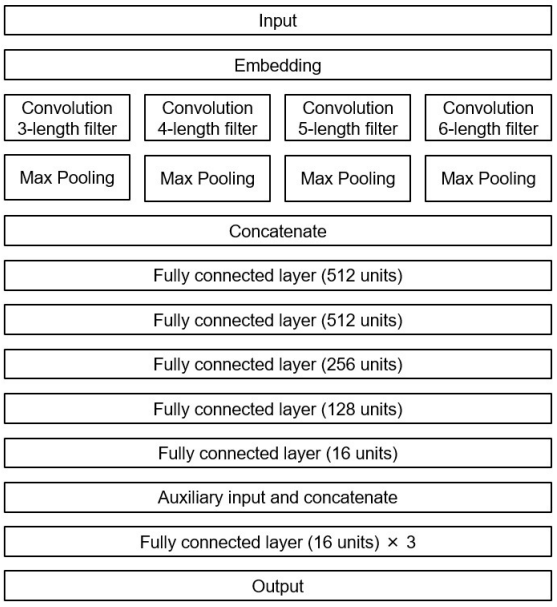


図 3 文字列とドメインの有効期間を特徴量としたモデル

$k = 32$ とした．畳み込み層では，4 種類の異なるウィンドウサイズのフィルタによって特徴を抽出する．ここで，ウィンドウサイズは $h = 3 \sim 6$ とし，各フィルタは 256 枚ずつ用意した．プーリング層では，各フィルタによって生成された特徴マップから最大値を抽出する．次に，各最大値を連結することで 1024 次元ベクトルを生成し，全結合層に入力する．全結合層における Dropout[5] の割合は 0.5 とした．提案モデルでは，特徴量のスケールを合わせるために Batch Normalization[6] を使用した．活性化関数は，畳み込み層と全結合層では ReLU，出力層ではシグモイド関数を用いた．

4. 性能評価

4.1 WarpDrive

Web 媒介型攻撃は特定の Web サイトを閲覧したユーザに対してのみ攻撃が行われるため，正確な実態把握が困難である．そこで，Web 媒介型攻撃の観測及び分析を目的としたプロジェクトが WarpDrive[7] である．WarpDrive では専用のアプリケーションを配布することで，ユーザが Web サイトを閲覧した際のアクセスログを収集している．アクセスログには URL やタイムスタンプ，リファラなどの情報が記録されている．また，収集された URL には Google Safe Browsing (GSB) [8] とのマッチング情報が付与されている．本研究では，GSB によるマッチング情報をラベルとして，WarpDrive で収集された URL を利用した．

表 2 評価用データセットの内訳

	悪性 URL	良性 URL
WarpDrive	389	11834
PhishTank	3801	0
OpenPhish	715	0

表 3 WarpDrive で収集された悪性 URL の月ごとの内訳

月	4	5	6	7	8	9	10	11
件数	30	26	28	210	23	25	21	26

4.2 評価用データセット

本実験で用いたデータセットの内訳を表 2 に示す．悪性 URL には，WarpDrive[7] で 2019 年 4 月〜11 月までに収集された URL を用いた．表 3 に WarpDrive で収集された悪性 URL の月ごとの内訳を示す．それに加え，インターネット上でフィッシングサイトの情報を公開している PhishTank[9]，OpenPhish[10] から取得した URL も悪性 URL として利用した．良性 URL には，WarpDrive で 2019 年 4 月 1 日に収集された URL を用いた．なお，データの偏りを防ぐために，悪性 URL と良性 URL はそれぞれ FQDN の重複が無いものを使用した．これにより，モデルが FQDN のみに注目して予測を行うことを防止する．また，入力文字列の長さは先述の通り，200 文字で固定とした．200 文字より長い場合は超過部分を削除し，短い場合は 200 文字になるようにパディングを行った．ドメインの有効期間は，UNIX 時間に変換した登録年月日と有効期限の差とした．

4.3 実験結果

10 分割交差検証を行った結果を表 4 に示す．各指標の値は交差検証で得られた結果の平均値である．表 4 より，文字列とドメインの有効期間を特徴量としたモデルが全ての指標で優れていることがわかる．

5. グレーリストの作成

WarpDrive で収集された URL には，GSB に登録されていない悪性 URL が含まれている可能性がある．しかし，

表 4　10 分割交差検証の結果

特徴量	正解率	適合率	再現率	F 値
文字列のみ	0.944	0.967	0.924	0.944
文字列・有効期間	0.952	0.974	0.930	0.951

表 5　学習データセットの構成

	悪性 URL	良性 URL
WarpDrive	755	11834
PhishTank	6877	0
OpenPhish	2390	0

膨大な量のデータから悪性 URL を発見するには多大な労力と時間を要する．そこで，提案モデルを用いて，大規模分析基盤から悪性度の高い URL のリストであるグレーリストを作成する実験を行った．本実験では，WarpDrive によるデータ収集時点では GSB に登録されていなかったが，後に GSB に登録された URL の検知を目的とした．

5.1　学習用データセット

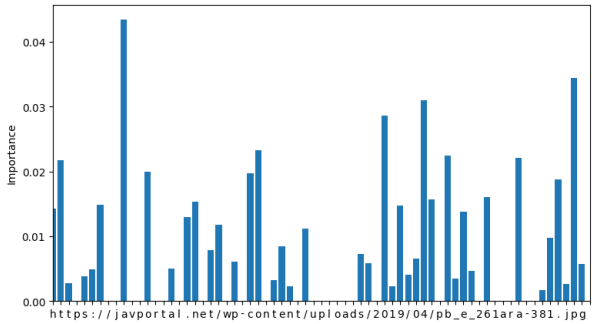
Character-level CNN モデルの学習に用いたデータセットの内訳を表 5 に示す．悪性 URL には，WarpDrive で 2019 年 4 月に収集された URL と，PhishTank，OpenPhish から取得した URL を用いた．訓練用の悪性 URL については，より多くのデータを用いて学習を行うために FQDN の重複を認めて利用した．良性 URL には，WarpDrive で 2019 年 4 月 1 日に収集された URL を用いた．

5.2　Google Safe Browsing

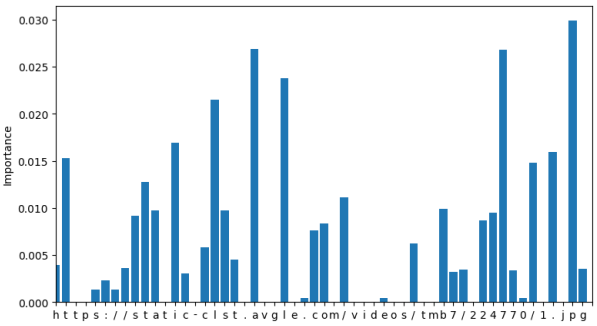
Google Safe Browsing (GSB) は悪性 URL に含まれるパターンを SHA-256 によるハッシュ値に変換してブラックリストに登録している．URL が悪性であるか判定する際は，部分文字列を抽出し，それらのハッシュ値をブラックリストと照合する．また，悪性 URL にはソーシャルエンジニアリングやマルウェアなどの脅威タイプが設定されている．さらに，脅威タイプがマルウェアであるものはマルウェア入口サイトとマルウェア配布サイトに分類される．多くの場合，マルウェア入口サイトは iframe タグやスクリプトによって，実際に攻撃を行う悪性サイトからコンテンツをロードするサイトである．また，マルウェア配布サイトはユーザの PC にマルウェアをダウンロードさせるサイトである．本実験では，2017 年 3 月〜2019 年 12 月に登録されたハッシュ値を用いて，GSB による悪性判定の時間的な変化を調査した．

5.3　実験結果

WarpDrive で 2019 年 4 月 30 日に収集された際に，GSB に登録されていなかった URL11320 件を調査したところ，そのうち 26 件が後に GSB に登録されていた．この URL



(a) モデルが検知できた URL



(b) モデルが検知できなかった URL

図 4　Grad-CAM による予測根拠の可視化: (a) モデルが検知できた URL，(b) モデルが検知できなかった URL

群に対して，学習済みのモデルを用いて予測を行ったところ，2725 件が悪性度の高い URL であると判定された．そのうち，後に GSB に登録された URL は 20 件であった．元の URL 群とグレーリストにおける GSB による判定結果を表 6 に示す．全 URL に対する悪性 URL の割合を r とする．元の URL 群では，

$$r = \frac{26}{11346} \simeq 0.002 \tag{4}$$

であるのに対して，グレーリストは

$$r = \frac{20}{2745} \simeq 0.007 \tag{5}$$

となっており，グレーリストの方がより多くの悪性 URL を含んでいた．また，後に GSB に登録された URL における脅威タイプの内訳を表 7 に示す．ソーシャルエンジニアリングは 75%，マルウェア入口サイトは 80%を検知し，元の URL 群に 1 件存在したマルウェア配布サイトについても検知できた．また，全体の再現率は 0.769 であった．

5.4　予測根拠の可視化

モデルが検知できた，脅威タイプがマルウェア入口サイトである URL のうち，3 件の URL が共通の部分文字列 “jav” を含んでいた．Grad-CAM[11] を用いて Character-level CNN における予測の根拠を可視化した結果を図 4 に示す．横軸は URL 文字列，縦軸はモデルの予測に影響を与えた度合い，すなわち重要度を表す．Grad-CAM は畳み込みによって生成された特徴マップから算出されるため，グラフ

表 6　GSB による判定結果

	GSB に登録されなかった URL	後に GSB に登録された URL	合計
元の URL 群	11320	26	11346
グレーリスト	2725	20	2745

表 7　後に GSB に登録された URL の脅威タイプ

	ソーシャルエンジニアリング	マルウェア入口サイト	マルウェア配布サイト	合計
元の URL 群	20	5	1	26
グレーリスト	15	4	1	20

中の各文字に対応する重要度は隣接する文字の情報を含んでいる．図 4(a) の URL は，部分文字列 “javportal.net/” が悪性なパターンとして GSB に登録されており，モデルは “jav”の辺りに注目して予測を行っていることがわかる．また，モデルの学習に用いた悪性 URL のうち，129 件が “jav” を含んでいた．このことから，モデルが訓練データから悪性なパターンを学習して，予測を行ったと考えられる．一方，図 4(b) はモデルが検知できなかった URL の Grad-CAM である．この URL は “avgle.com/videos/” の部分が悪性なパターンとして GSB に登録されている．モデルは “avgle”の辺りに注目できているが，良性であると判定した．これはモデルの学習に用いた良性 URL のうち，14 件に “avgle”が含まれていたが，悪性 URL には含まれていなかったことが原因であると考えられる．

6. おわりに

本研究では，Character-level CNN で抽出した文字列に関する情報と，ドメインの有効期間を特徴量とした悪性 URL 検知モデルを提案した．実在する悪性サイトの URL を用いた実験の結果，提案モデルは文字列のみを特徴量としたモデルと比べて，正解率，適合率，再現率，F 値がそれぞれ 1%程度上昇した．また，提案モデルを用いて大規模分析基盤に保存されている膨大な量の URL から，悪性の可能性のある URL のリストであるグレーリストを作成した．その結果，WarpDrive によるデータ収集時点では GSB に登録されていなかったが，後に GSB に登録された URL26 件の内，20 件を検知することができた．その際，約 2700 件の誤検知があったが，元の URL 群と比較してグレーリストでは，全 URL に対する悪性 URL の割合が増加した．そのため提案モデルは，より詳細な解析を行う前に膨大なデータから悪性サイトを絞り込む用途での使用を期待できる．

今後の課題として，ドメインに関する情報の取得方法の改善が挙げられる．WHOIS プロトコルは，レジストリ間で応答の形式が統一されていないため，コンピュータによる処理が簡単ではない．これに対して，WHOIS の次世代プロトコルである Registration Data Access Protocol (RDAP) は応答の形式が JSON で統一されているため，コンピュータによる処理が容易である．しかし，現状 RDAP は一部のドメインにしか対応していない．RDAP が十分に整備されれば，悪性 URL 検知モデルの特徴量として利用できる情報の取得が容易になると考えられる．今後の展望として，悪性 JavaScript 検知モデルなどの異なるモデルと組み合わせることで，より頑強な悪性サイト検知システムの構築が期待できる．

謝辞　本研究成果は，国立研究開発法人情報通信研究機構（NICT）の委託研究「Web 媒介型攻撃対策技術の実用化に向けた研究開発（通称 WarpDrive）」により得られたものです．本研究を実施するにあたり，WarpDrive プロジェクトのメンバーには様々な有益な助言を頂きました．ここに深甚なる謝意を表します．

参考文献

[1] Bilge, L., Kirda, E., Kruegel, C. and Balduzzi, M.: EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis., *Ndss*, pp. 1–17 (2011).
[2] Saxe, J. and Berlin, K.: eXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys, *arXiv preprint arXiv:1702.08568* (2017).
[3] Sahoo, D., Liu, C. and Hoi, S. C.: Malicious URL detection using machine learning: A survey, *arXiv preprint arXiv:1701.07179* (2017).
[4] Zhang, X., Zhao, J. and LeCun, Y.: Character-level convolutional networks for text classification, *Advances in neural information processing systems*, pp. 649–657 (2015).
[5] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting, *The journal of machine learning research*, Vol. 15, No. 1, pp. 1929–1958 (2014).
[6] Ioffe, S. and Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift, *arXiv preprint arXiv:1502.03167* (2015).
[7] WarpDrive. `https://warpdrive-project.jp/`.
[8] Google Safe Browsing. `https://safebrowsing.google.com/`.
[9] PhishTank. `https://www.phishtank.com/`.
[10] OpenPhish. `https://openphish.com/`.

[11] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization, *Proceedings of the IEEE international conference on computer vision*, pp. 618–626 (2017).