CL-007

ハッシュ関数の特徴を用いた深層学習によるマルウェア分類

Malware Classification by Deep Learning Using the Characteristics of Hash Functions

馬場 隆寛 † 馬場 謙介 ‡ 山内 利宏 §
Takahiro Baba Kensuke Baba Toshihiro Yamauchi

1. はじめに

インターネットの発展に伴い，Iot 機器が増えている．それとともに IoT 機器がマルウェアに感染するリスクも増えており，マルウェアを検知することは，サイバーセキュリティの分野において重要な課題である．

近年，表層解析ログを用いたマルウェアの検知の研究が行われており，データの収集が盛んになりつつある．動的解析ログでは実際にマルウェアを動作させてログを収集するためマルウェアの検知に有用な情報であり，研究 [8] が行われてきた．動的解析ログは，一度マルウェアを実行してログを収集しなければならず，実行環境の準備する必要があり，大量にデータを集めるには時間がかかる．どういった実行環境を作って収集するかによって，外的要因を受けてしまうため，複数環境で収集されたログの場合，実行環境を考慮しなければならない．そこで，表層解析ログを用いてマルウェアを検知できれば，前述の問題を解決できる．

また，計算機の処理能力の向上により，機械学習 [6] が成果を挙げている．中でもラベルの付与された大規模なデータに対して，深層学習 [15] を適用することは，有効な手段であり様々な研究 [7] が行われている．機械学習では，分析するデータに対して，特徴量を人手で抽出する必要があった．この特徴量の選び方によって，分類性能に大きな影響がある．これに対して，深層学習では，特徴量を自動的に抽出できるという利点がある．

そこで，本研究では，マルウェアの表層解析ログに対して，深層学習を適用することにより，マルウェアと非マルウェアの分類を行った．本研究では，FFRI Dataset 2020 [18] にて提供されている表層解析ログを使用して評価実験を行った．まず，PE 表層情報に対して，多層パーセプトロン [11] を使用して学習を行い，深層学習がマルウェア検知に有効であることを検証した．また，文字列であるハッシュ値に対して，Character-level CNN [13] を用いてマルウェア分類を行った．学習の性能評価として，分類精度，適合率，再現率の値を算出することで，数種類のハッシュ値のうち特にマルウェア検知に有効なハッシュ値について検証を行った．

†岡山大学 学術研究院自然科学学域
‡岡山大学 サイバーフィジカル情報応用研究コア
§岡山大学 学術研究院自然科学学域/JST さきがけ

2. 関連研究

2.1. 深層学習に関する研究

深層学習は，大量のデータが利用可能な場合，特徴量を自動で抽出することができる．サイバーセキュリティの研究において，マルウェアの検出や分類の問題は，大規模なデータを扱っており，様々な研究 [12] が行われている．

また，表層解析ログを用いてニューラルネットワークでマルウェアを検知する研究 [19] が行われており，True positive rate の改善に効果でている．

表層解析ログを用いた研究 [20] では，PE 表層情報とハッシュ値に対して，機械学習を適用し，静的なマルウェア検知を行っている．PE 表層情報に対して，従来の機械学習を用いて学習させている．また，4 種類のハッシュ値を使用して学習を行っており，マルウェアの分類に有効かどうか検証を行っている．しかし，ハッシュ値はいくつもの種類があり，そのうち 4 種類のハッシュを用いている理由が明確ではなかった．

このため，本研究では，学習法として深層学習を用いて PE 表層情報使用した分類実験を行った．また，ハッシュ値に関してもそれぞれのハッシュ値ごとの分類性能の比較を行った．

2.2. ハッシュ値を用いた研究

fuzzy hash [9]，peHash [14] といったハッシュ値を計算する手法を用いることで，PE 表層情報よりもマルウェアの分析を高速化できる．特にデータサイズが大きい場合には，文字列であるハッシュ値で分類できれ

ば，素早くマルウェアを検知することにつながる．しかし，マルウェア検知に関して，どのハッシュ値が深層学習に有用か検証した研究は行われていない．

そこで，本研究では，各ハッシュ値がどの程度，有用か検証を行った．まず，fuzzy hash とは，部分文字列をずらしつつハッシュ化する手法と部分文字列をハッシュ化して結合する手法を組み合わせた手法である．fuzzy hash では，ファイルや文章が類似している場合には類似した値になるハッシュ値となる．本研究で使用しているデータには impfuzzy [1]，ssdeep [4]，TLSH [5] が含まれている．impfuzzy は import API からハッシュ値を計算している．ssdeep では，ファイルの先頭から順に文字列を生成していくハッシュ化手法である．tlsh は Linux の実行ファイル形式である，ELF ファイル形式からハッシュ値の計算を行っている．

次に，peHash とは，マルウェアのクラスタリングを行うために考えられたハッシュ値である．マルウェアの開発環境が同じ場合などにハッシュ値が同じになるようになっている．これらのハッシュ値を使いマルウェアの種類の分類する研究 [10] が行われている．本研究で使用しているデータには，pehash [3] を用いて収集された，imphash，totalhash，EndGame，AnyMaster，Crits，peHashNG が含まれている．これらのハッシュ値は PE ファイルや import API から md5，sha1 といったハッシュ関数を用いて計算されている．

3. マルウェア分類手法

3.1. データセット

1 章で述べたように表層解析ログを用いることは，リスクなどの観点から動的解析ログを用いる場合と比較して，利点がある．そこで本研究では，表層解析ログのデータセットとして，FFRI Dataset 2020 を用いて実験を行った．FFRI Dataset 2020 は，MWS Dataset 2020 の一部として提供されているデータである．マルウェア 75,000 件，クリーンウェア 75,000 件のハッシュ値および，PE 表層情報が提供されている．このうちそれぞれ，9 割にあたる 67,500 件ずつを訓練データとし，残りの 1 割にあたる 7,500 件ずつをテストデータして評価を行った．

FFRI Dataset 2018 を使った研究 [20] では，テストデータのマルウェアの数とクリーンウェアの数の間で，均衡が取れていないため，見かけ上の分類精度が良くなるという懸念がある．このため，本研究では，マルウェア，クリーンウェアともに 7,500 件をテストデータとし，均衡のとれたテストデータとしている．

3.2. PE 表層情報を用いた分類手法

一つ目の実験として，PE 表層情報を用いたマルウェアの分類を行った．PE 表層情報は，pefile [2] から得られたダンプファイルから抽出した情報である．PE 表層情報のデータ内容は，表 1 のようになっている．学習方法は，深層学習の一種である多層パーセプトロンを用いている．多層パーセプトロンには，利点として訓練データを使って重みを自動調整できる点と，どのような学習データに対しても学習を行える点がある．他の深層学習の手法は，画像処理向きや自然言語処理向きというように向き不向きがある．このため，PE 表層情報を用いた分類では，多層パーセプトロンを用いた．

表 1: PE 表層情報のデータ内容

フィールド名	内容
PE	32bit または 64bit
DLL	DLL か否か
Packed	パッキングの有無
Anti-Debug	AntiDebug の有無
GUI Program	GUI プログラムか否か
Console Program	Console プログラムか否か
mutex	mutex の有無
contains base64	Base64 文字列の有無
PEiD	マッチした PEiD シグネチャ名
AntiDebug	AntiDebug 手法

3.3. ハッシュ値を用いた分類手法

ハッシュ値を用いた分類では，学習方法として，畳み込みニューラルネットワークを文字レベルで適用した Character-level CNN [13] を用いている．Character-level CNN は畳み込みを行う際に 1 文字ごとの前後のつながりの情報と合わせて畳み込みを行うことができる．今回使用しているハッシュ値はファイルが類似した場合には似た文字列となるため Character-level CNN は有効な手段だと想定できる．

4. 実験結果

4.1. PE 表層情報を用いたマルウェア検知

ベクトル化を行うにあたり，図 1 のように PE 表層情報のそれぞれのフィールドの値をすべて調査し，その値の有無によりバイナリベクトルとしている．この

ようなベクトルにしているのは，“yes”,“no”の 2 値ではないフィールドが多いためである．例えば，DLL の値には “yes”,“no”,“yes(no)”,“no(yes)” という 4 パターンの文字列がある．このため，DLL が “yes” の場合は，“[1,0,0,0]” というベクトルとしている．こられのベクトルをフィールドごとに求め，順番に結合させて，全体のベクトルとしている．

3.2 節で記述した手法により，実験を行った結果，表 2 のような結果となった．分類精度，適合率，再現率ともに高い性能となっている．

PE	32 bit
DLL	no
Packed	no
Anti-Debug	no
GUI Program	yes
Console Program	yes
Mutex	no
Contains base64	yes
PEiD	['Microsoft_Visual_Cpp_v50v60_MFC']
AntiDebug	[ ]

↓

[1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

図 1: PE 表層情報のベクトル化の例

表 2: PE 表層情報を用いた学習の評価

	分類精度	適合率	再現率
PE 表層情報	0.9637	0.9628	0.9652

4.2. ハッシュ値を用いたマルウェア検知

二つ目の実験として，ハッシュ値を用いたマルウェア検知を行った．FFRI Dataset 2020 で提供されているハッシュ値のうち，fuzzy hash である impfuzzy，ssdeep，TLSH を用いている．また，fuzzy hash とは別に peHash である，imphash，totalhash，endgame，anymaster，crits，pehashng も用いている．これら全 9 種類のハッシュ値について，比較を行った．また，9 種類すべてのハッシュ値を組み合わせた値での実験を行った．機械学習を行うにあたり，図 2 のようにハッシュ値の文字を整数に対応させ，ベクトル化を行った．実験の結果，表 3 のような結果となった．impfuzzy を用いた場合が分類精度，適合率，再現率ともに最も高い性能となった．

48:tapqG9O6gs9V8+tmC6U0nB/

↓

[52, 56, 58, 116, 97, 112, 113, 71, 57, 79, 54, 103, 115, 57, 86, 56, 43, 116, 109, 67, 54, 85, 48, 110, 66, 47]

図 2: ハッシュ値のベクトル化の例

表 3: ハッシュ値ごとの学習の評価

	分類精度	適合率	再現率
fuzzy hash			
impfuzzy	0.8500	0.8502	0.8497
ssdeep	0.6402	0.6388	0.6424
TLSH	0.8035	0.8033	0.8035
peHash			
imphash	0.7947	0.7963	0.7937
totalhash	0.7285	0.7280	0.7287
endgame	0.7562	0.7556	0.7577
anymaster	0.7742	0.7742	0.7742
crits	0.7360	0.7363	0.7372
pehashng	0.7725	0.7722	0.7722
全ハッシュ	0.8259	0.8263	0.8257

5. 考察

PE 表層情報を用いた実験では，分類精度，適合率，再現率ともに高い性能を得ることができた．ハッシュ値を使った実験では，impfuzzy を使用した場合の性能が最も良く，9 種類のハッシュ値すべてを組み合わせた場合よりも良い性能となった．ハッシュ値を用いてマルウェアを検知することは可能だが，ハッシュ値単体では，PE 表層情報を利用した場合の性能を超えることはできなかった．茂木の研究 [20] では，PE 表層情報

と 4 種のハッシュ値を組み合わせることでより高い分類精度を実現しているが，本研究での実験結果を踏まえると impfuzzy のみ PE 表層情報と組み合わせるだけで十分に高い分類精度，もしくはより高い分類精度が得られる可能性がある．

6. まとめ

6.1. まとめ

本研究では，PE 表層情報とハッシュ値それぞれに対して，深層学習を適用し評価を行った．その結果，PE 表層情報を使用した場合は高い性能が得られた．また，ハッシュ値それぞれを使用した場合は，impfuzzy を使った場合の性能が最もよく，全ハッシュ値を組み合わせて使用した場合よりも高い性能となった．これらの結果から表層解析ログに対して，深層学習は有効な学習方法であると言える．

6.2. 今後の課題

本研究では，まだ層学習のハイパーパラメーターの調整を行っていないため，より高い性能を得るには，パラメーターチューニングを行う必要がある．

本研究では，FFRI Dataset 2020 のデータを用いており，合計 15 万件ほどのデータである．FFRI Dataset 2018 [17]， FFRI Dataset 2019 [16] でも同様の表層解析ログが提供されている．組み合わせて使用することで 80 万件ほどのデータを使用することが可能であり，より大規模なデータとなるため学習の性能の向上が期待できる．

謝辞 本研究の一部は，JST さきがけ JPMJPR1938, および JSPS 科研費 JP19H05579 の助成を受けたものです．

参考文献

[1] : Import API と Fuzzy Hashing でマルウエアを分類する 〜impfuzzy〜(2016-05-09)，https://blogs.jpcert.or.jp/ja/2016/05/impfuzzy.html. Accessed Jun. 8, 2021.

[2] : pefile, https://github.com/erocarrera/pefile. Accessed Jun. 8, 2021.

[3] : pehash, http://github.com/knowmalware/pehash. Accessed Jun. 8, 2021.

[4] : ssdeep, https://ssdeep-project.github.io/ssdeep/index.html.

[5] : Trend Micro Locality Sensitive Hash, https://github.com/trendmicro/tlsh. Accessed Jun. 8, 2021.

[6] Bishop, C. M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag, Berlin, Heidelberg (2006).

[7] Dargan, S., Kumar, M., Ayyagari, M. R. and Kumar, G., A survey of deep learning and its applications: A new paradigm to machine learning, *Archives of Computational Methods in Engineering*, pp. 1–22 (2019).

[8] Kawaguchi, N. and Omote, K., Malware function classification using apis in initial behavior, *2015 10th Asia Joint Conference on Information Security*, IEEE, pp. 138–144 (2015).

[9] Kornblum, J., Identifying almost identical files using context triggered piecewise hashing, *Digital Investigation*, Vol. 3, pp. 91–97 (2006).

[10] Li, Y., Sundaramurthy, S. C., Bardas, A. G., Ou, X., Caragea, D., Hu, X. and Jang, J., Experimental Study of Fuzzy Hashing in Malware Clustering Analysis, *8th Workshop on Cyber Security Experimentation and Test (CSET 15)*, Washington, D.C., *USENIX* Association (2015).

[11] Noriega, L., Multilayer perceptron tutorial, *School of Computing. Staffordshire University* (2005).

[12] Qiu, J., Zhang, J., Luo, W., Pan, L., Nepal, S. and Xiang, Y., A Survey of Android Malware Detection with Deep Neural Models, *ACM Comput. Surv.*, Vol. 53, No. 6 (2020).

[13] Saxe, J. and Berlin, K., eXpose: A Character-Level Convolutional Neural Network with Embeddings For Detecting Malicious URLs, File Paths and Registry Keys, *CoRR*, Vol. abs/1702.08568 (2017).

[14] Wicherski, G., peHash: A Novel Approach to Fast Malware Clustering, *2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET 09)*, Boston, MA, USENIX Association (2009).

[15] 岡谷貴之，ディープラーニング，映像情報メディア学会誌， Vol. 68, No. 6, pp. 466–471 (2014).

[16] 荒木粧子，笠間貴弘，押場博光，千葉大紀，畑田充弘，寺田真敏，マルウェア対策のための研究用データセット〜MWS Datasets 2019〜，情報処理学会研究報告， Vol. 2019-SPT-34, No. 8, pp. 1 – 8 (2019).

[17] 高田雄太，寺田真敏，松木隆宏，笠間貴弘，荒木粧子，畑田充弘，マルウェア対策のための研究用データセット〜MWS Datasets 2018〜，情報処理学会研究報告， Vol. 2018-SPT-29, No. 38, pp. 1 – 8 (2018).

[18] 寺田真敏，秋山満昭，松木隆宏，畑田充弘，篠田陽一，マルウェア対策のための研究用データセット MWS Datasets〜コミュニティへの貢献とその課題〜，情報処理学会研究報告，Vol. 2020-IFAT-139, No. 8, pp. 1 – 6 (2020).

[19] 樋川卓也，表層解析データを入力とするアンサンブルニューラルネットワークを用いたマルウェア検知，コンピュータセキュリティシンポジウム 2020 論文集，pp. 632–636 (2020).

[20] 茂木裕貴，PE 表層情報を用いた機械学習による静的マルウェア検知，2019 年暗号と情報セキュリティシンポジウム (SCIS 2019) 論文集 (2019).