Computer Security Symposium 2017
23 - 25 October 2017

多種環境におけるマルウェア解析レポートを用いたマルウェア分類手法に関する一検討

藤井翔太[†1]　鬼頭哲郎[†1]　重本倫宏[†1]　藤井康広[†1]

概要：近年，マルウェアが高度化・膨大化しており，その脅威が年々増加している．このような状況において，マルウェア解析の更なる自動化・効率化が求められている．そこで，多種解析環境での解析結果を用いたマルウェア分類手法を提案する．同手法では，多種解析環境でのマルウェア自動解析レポートから時系列データ・非時系列データを抽出し，深層学習によりそれぞれを並列に学習後，両者を結合する形で分類器を構築する．また，同分類器を用いて解析対象を既知のマルウェアファミリに分類する．これにより，人手での解析前にその挙動を推定し，解析の効率化を図ることができる．本稿では，提案手法のプロトタイプを実装し，その性能を評価した結果を報告する．

キーワード：マルウェア，分類，機械学習，深層学習

Proposal of Method for Malware Classification using Malware Analysis Report of Multiple Analysis Environment

SHOTA FUJII[†1]　TETSURO KITO[†1]
TOMOHIRO SHIGEMOTO[†1]　YASUHIRO FUJII[†1]

Abstract: Recently, impacts and threats caused by cyber-attacks using malware are increasing. In such situation, the further improvement in the efficiency of malware analysis is required. To mitigate the issue, we propose an automatic malware classification method using reports from a malware dynamic analysis system with multiple runtime environments. The proposed method classifies malware into known malware families using deep learning techniques with timeseries/non-timeseries data. By this method, malware analysts can estimate behavior of malware before starting manual analysis. Consequently, we expect the efficiency of malware analysis will increase. In this report, we describe a design of the proposed method. We also report the evaluation result with the prototype.

Keywords: malware, classification, machine learning, deep learning

1. はじめに

近年，サイバー攻撃に用いられるマルウェアが高度化・膨大化しており，その脅威は年々増加している．ここで，従来のマルウェア対策手法の一つに，シグネチャベースのものがある．しかし，前述のようにマルウェアが高度化・膨大化している現状から，そのすべてに対応するのは難しくなってきており，従来型の対策手法は，全体の約 45%しか検出できないという報告もある[1]．このような状況においては，次々と発生するマルウェアの挙動を解明し，セキュリティ対策に繋げることが重要となってきているものの，現状では専門家の手動解析に依る部分が依然として残されており，膨大なマルウェアすべてに対応することは難しい．

このような背景から，様々な自動解析技術が研究開発されている．同技術を用いることによって，膨大なプログラムの中からより不審なものを抽出することが可能になる．これにより，専門家による手動解析の対象をより不審なものに絞り込むことができ，解析効率を向上することが可能となった．一方で，自動解析においても，完全な機械化は難しく，詳細の確認や最終的な判断など，一部は人手に依る部分もある．前述のようなマルウェアが膨大化しているという背景もあり，人手に依る部分をできるだけ削減し，解析をより効率化することが求められている．

ここで，マルウェアが膨大化している原因のひとつに，亜種マルウェアの存在がある．亜種マルウェアとは，既知のマルウェアと同様のプログラムを利用していたり，挙動が類似していたりするもののことである．このことから，解析対象を既知のマルウェアファミリに分類することによって，そのベースとなる挙動が既に解析されており，専門家による手動解析が不要な亜種マルウェアを事前に選別し，解析効率を向上することが期待できる．また，いずれのマルウェアファミリにも分類できないものをより詳細に手動解析することが望ましい新種のマルウェアとして抽出することも期待できる．

そこで，本報告では，多種解析環境での解析結果を基にマルウェアを分類し，人手での解析を補助する手法を提案する．同手法は，解析環境間での動作の違いや API コール列等を特徴量とし，深層学習によって解析対象を既知のマ

†1 株式会社日立製作所
Hitachi Ltd.

ルウェアファミリに分類する．これにより，前述のような解析対象の既知マルウェアファミリへの分類や新種マルウェアの抽出，および，これらを通した解析全体のさらなる効率化が期待できる．本報告では，提案手法の設計を述べる．また，プロトタイプを実装し，データセットを用いてその性能を評価する．

2. 関連研究と本研究の解決したい課題

2.1 関連研究

(1) マルウェア解析環境

ここでは，マルウェア解析環境を用いて動的解析を行う既存手法について述べる．マルウェア解析環境を用いた動的解析については，多くの研究がなされており，Anubis[2]，Ether[3]，CWSandbox[4]，TTAnalyze[5]，Panorama[6]等がある．ただし，前述したように，近年のマルウェアは高度化が進んでおり，こうした動的解析の環境を検知してその動作を止めるような解析妨害機能を有しているものもある．実際に文献[7]において，Anubis は 84.78%，Ether は 78.18%の解析環境検知型のマルウェアに回避されてしまうといった報告もされている．このような状況から，解析妨害機能を回避することが重要となっている．

ここで，解析妨害機能を回避するアプローチとして，複数の解析環境を用意することによってマルウェアの動作発現確率の向上を図るものがある．本アプローチを採用している手法の 1 つに，我々のグループで研究開発を行っている M3AS[8]がある．M3AS は，マルウェアを多種類の解析環境で実行させることで環境を選ぶマルウェアをも自動的に解析し，その挙動や動作環境を推定してレポートする．また，複数の解析環境を用いるものとしては，BareCloud[7]も挙げられる．BareCloud は，複数の著名な解析環境と非解析環境に近づけた環境の両方を用意し，各環境で同じ検体を動作させた際に，その挙動に一定の差が見られた場合には，解析環境検知型のマルウェアであるとして検出する．

(2) マルウェア推定手法

ここでは，マルウェア推定手法について述べる．ここでのマルウェア推定手法とは，解析対象がマルウェアか否かを推定する手法のことであり，大きく静的解析による手法と動的解析による手法の 2 つがある．静的解析による手法の 1 つに，PE-Miner[9]がある．PE-Miner は，PE ヘッダの n-gram を特徴量に，決定木等のアルゴリズムを用いて解析対象がマルウェアか否か推定する手法である．また，文献[10]では，バイトコードの n-gram を特徴量とし，ナイーブベイズやサポートベクターマシン（SVM）等によって解析対象がマルウェアか否か推定している．動的解析による手法には，文献[11]のものがある．文献[11]は，解析対象を動的解析し，API Monitor[12]を利用して取得した API の引数や呼び出し列を基に，各種機械学習アルゴリズムによって解析対象がマルウェアか否かを判定するものである．また，文献[13]においても同様に，API 呼び出し列を用いたマルウェア推定手法が提案されている．

しかし，マルウェア推定に静的解析を用いる場合，様々な解析妨害機能が施されている場合があり，そのすべてを回避しつつ解析を実施することは難しく，汎用性にかける部分がある．例えば，マルウェア本体に，パッキングが施されている場合があり，その際にはアンパックを行い，マルウェア本体を抽出する必要がある．このパッキングには，独自のパッカーが利用されている場合も少なくなく，アンパックは容易ではない．また，動的解析を用いる際，マルウェアの解析妨害機能によってその挙動が発現しなかった場合は，適切に推定できない恐れがある．

(3) マルウェア分類手法

ここでは，マルウェア分類手法について述べる．ここでのマルウェア分類手法とは，解析対象のマルウェアを既知のマルウェアファミリに分類するもの，あるいは類似の特徴を有するマルウェア同士を同じグループにクラスタリングするものである．また，(2)のマルウェア推定手法と同様に，静的解析によるものと動的解析によるものの 2 つに大別できる．静的解析によるものには，文献[14]の手法がある．文献[14]は，API 推移依存グラフの類似度を Dice 係数により算出し，階層型クラスタリングでマルウェアの亜種グループを抽出している．動的解析による手法としては，文献[15]のものがある．文献[15]は，システムコールの呼び出しログの引数から，依存グラフを作成するとともに，グラフの類似度を算出することによって，亜種マルウェアを検出している．また，文献[16]は，http，smtp，udp，および tcp 等の各種ネットワーク挙動の特徴を捉え，解析対象のマルウェアをマルウェアファミリに分類する手法を提案している．

しかし，(2)のマルウェア推定手法と同様，静的解析の場合は，汎用性にかける，動的解析の場合は，挙動が発現せずに適切な分類ができない恐れがあるといった課題がある．

2.2 本研究の解決したい課題

上述したように，マルウェア分類手法を用いて，未知のマルウェアを既知のマルウェアファミリに分類することによって，解析前にその挙動を推定でき，解析の助けとなる．また，新種のマルウェアを抽出することも期待できる．ここで，静的解析によるマルウェア分類は汎用性にかける部分があるために，動的解析を用いた分類を行いたい一方で，解析妨害機能によって動作が発現しない場合があり，その際には適切な分類ができない課題がある．

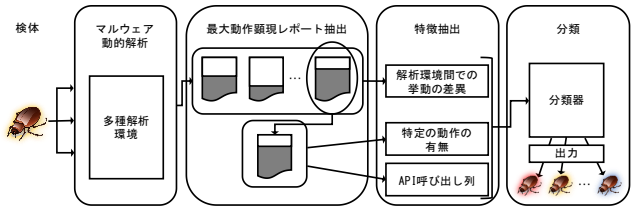


図 1 提案手法の全体像

3. 提案手法

3.1 課題の解決方法

2.2 節で述べたように，マルウェアの分類手法には，静的解析によるものと動的手法によるものがあるが，それぞれに汎用性にかける部分がある，マルウェアの動作が発現しない場合があるといった課題がある．ここで，動的解析においては，2.1 節で述べたように，複数解析環境を用いることでマルウェアの発現可能性を向上することができる．これにより，動的解析によるマルウェア分類の課題を緩和することが期待される．

そこで，本研究では，複数解析環境におけるマルウェアの動的解析レポートを活用したマルウェア分類手法について検討する．以降の節では，同手法の詳細について述べる．

3.2 提案手法の概要

提案手法は，3.1 節で述べたように，複数解析環境におけるマルウェアの解析レポートを活用し，マルウェアを分類するものである．ここで，提案手法の全体像を図 1 に示し，各構成要素について以下で述べる．

(1) マルウェア動的解析

解析対象のマルウェアを受け取り，動的解析を行った後，その解析レポートを出力する．なお，前述のように，提案手法は，多種解析環境における解析レポートを活用して解析対象のマルウェアを分類することから，本解析環境は，多種解析環境を前提とする．

(2) 最大動作発現レポートの抽出

多種解析環境から出力された解析レポートのうち，最も解析対象の動作が発現したものを抽出する．これは，最も動作が発現したということは，解析対象の本来の性質を最も表しているために，そのようなレポートは，解析対象の適切な分類に有用であると考えられるためである．

(3) 特徴抽出

マルウェアの解析レポート，および抽出した最大動作発現レポートから特徴を抽出し，(4)のマルウェア分類器に投入できる形にする．

(4) 分類

投入された特徴量を基に，解析対象を分類し，その結果を出力する．

(1)のマルウェア動的解析に用いる多種解析環境には，例えばM3AS のような既存の多種解析環境を利用するものとし，残る(2)～(4)について，3.3～3.5 節で詳述する．

3.3 最大動作レポートの抽出

3.1 節で述べたように，解析対象の本来の性質を可能な限り捉えるために，多種解析環境で動作させた中で，最も動作の発現したレポートを抽出する．ここで，マルウェアを含むプログラムの動作には，API の呼び出しが伴うことから，多くの動作を示す検体は，そうでない検体に比べて API の呼び出し数が多いと考えられる．このため，今回は，各解析レポートから，最も多くの API を呼び出したものを最大動作発現レポートとして抽出する．

3.4 特徴抽出

本節では，提案手法において，マルウェアを分類するために利用する特徴量について検討する．

3.4.1 API 呼び出し列

前述のとおり，マルウェアを含むプログラムの動作には，API が伴う．このため，API の出現頻度や出現順序がマルウェア毎の特徴を表していると考えられる．なお，API の呼び出し列をマルウェアの推定・分類に利用する研究は多くあり，その有用性が実証されている[11][13][14]ことから，本手法においても，採用する．また，各 API は，呼び出されたユニークな API 次元数のベクトルで，表現したい API に対応する 1 つの要素だけが 1 で他の要素は全て 0 である 1-hot ベクトル（1-of-K ベクトル）として表現する．

3.4.2 解析環境間での挙動の差異

解析環境検知型のマルウェアは，自身の動作環境が解析環境であると判断した場合，例えば，自身のプロセスの終了など，真の挙動とは異なる動作を行う．ここで，多種解析環境において，そのようなマルウェアを動作させた際，解析環境間で，真の挙動と解析妨害挙動が混在する可能性がある．この現象を捉えることにより，解析環境検知型のマルウェアを検出できる．実際に，文献[7][17]などにおいて，動作環境間での挙動の差異が環境検知型のマルウェア検出に有用なことが実証されている．この解析環境間での挙動の差異が，マルウェアの分類にも寄与するのではないかと考え，今回特徴量として採用した．具体的な特徴量を表 1 に示す．これらは，ファイル生成や外部との通信など，マルウェアの動作として現れ得る部分を検討し，選定した．なお，通番 1～11 の値は，解析環境ごとの解析レポート A が N 個存在する場合，以下の式 1 で算出する．

$$dif(A_1, A_2, \cdots, A_N) = 1 - \frac{|A_1 \cap A_2 \cap \ldots \cap A_N|}{|A_1 \cup A_2 \cup \ldots \cup A_N|} \qquad (式\ 1)$$

式 1 は，解析レポート間の共通項を全要素で割ることで導出される解析レポート間の類似度を 1 から減算することによって，解析レポート間の差異を算出している．また，解析環境間での挙動の差異が大きいほど 1 に，小さいほど 0 に近づく．

表 1 解析環境間での挙動の差異に着目した特徴量

通番	特徴量	値
1	生成されたファイルの差異	0-1
2	tcp 通信先の差異	0-1
3	udp 通信先の差異	0-1
4	http 通信先の差異	0-1
5	https 通信先の差異	0-1
6	irc 通信先の差異	0-1
7	dns リクエストの差異	0-1
8	参照した hosts の差異	0-1
9	操作した mutex の差異	0-1
10	操作したレジストリの差異	0-1
11	user-Agent の差異	0-1
12	API 呼び出し数の差異	0-1
13	プロセス数の差異	0-1

表 2 特定動作の有無に着目した特徴量

通番	特徴量	値
1	自身の削除有無	0/1
2	自身のコピー有無	0/1
3	他プロセスへのコードインジェクション有無	0/1
4	自身の自動起動登録有無	0/1
5	実行可能ファイルの生成有無	0/1
6	メールの送信有無	0/1
7	tcp 通信の有無	0/1
8	udp 通信の有無	0/1
9	http 通信の有無	0/1
10	https 通信の有無	0/1
11	irc 通信の有無	0/1
12	hosts ファイルの書き換え有無	0/1

また，通番 12，13 の値は以下の式 2 で算出する．

$$\mathrm{dif}(A_1, A_2, \cdots, A_N) = 1 - \frac{\min(A_1, A_2, \ldots, A_N)}{\mathrm{Max}(A_1, A_2, \ldots, A_N)} \qquad \text{(式 2)}$$

式 2 は，算出対象の全解析レポートにおける最小数を最大数で割ることで導出される解析レポート間の類似度を 1 から減算することによって，解析レポート間の差異を算出している．また，式 1 と同様に，解析環境間での挙動の差異が大きいほど 1 に，小さいほど 0 に近づく．

3.4.3 特定動作の有無

マルウェアごとの特徴を捉えるために，特定動作の有無を特徴量として利用する．今回利用する特徴量の一覧を表 2 に示す. なお, これらの特徴量は, 主に文献[16]を参考に, マルウェアに現れ得る動作を選定している．

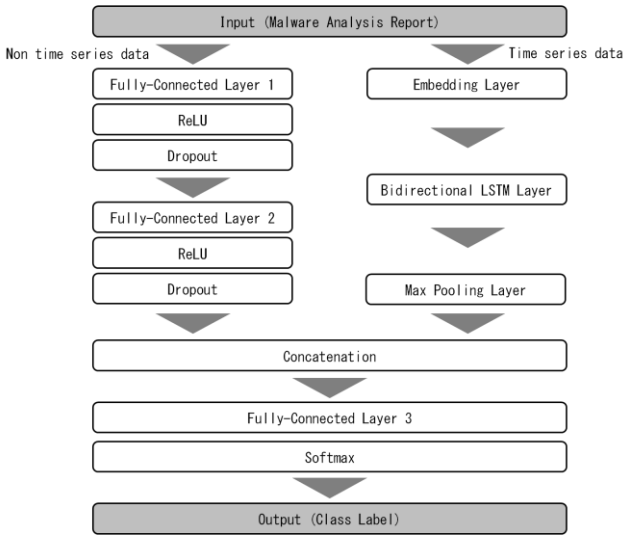


図 2 提案手法におけるネットワークの構成

3.5 分類

3.4 節で述べた特徴量を用いてマルウェアを分類する．ここで，API の呼び出し順序には，プログラムごとの特徴が反映されており[11][13][14]，時系列を保持したまま利用するのが望ましい．そこで，API 呼び出し列は，時系列データとして利用し，その分類には，系列データを高精度で学習可能な深層学習手法の 1 つである RNN（Recurrent Neural Network）を利用する．API 呼び出し列を時系列データとして利用することから，特徴量は，時系列のもの（API 呼び出し列）と非時系列のもの（解析環境間での挙動の差異，特定動作の有無）に大別できる．提案手法において，マルウェアを分類するために利用するネットワークを図 2 に示し，以降で説明する．

時系列のデータである API 呼び出し列に関しては, まず, 1-hot ベクトルを Embedding 層で分散表現にする．その後，Bidirectional LSTM（Long Short-Term Memory）層へ分散表現にした API 呼び出しの系列データを投入する. LSTM [18]は， RNN の一種であり，基本的な RNN よりも長期の記憶が可能な手法である．また，Bidirectional RNN[19]は，基本的な順向きの RNN に逆向きの RNN を加え，両者の出力を結合したものであり，順向きのみの場合よりも性能が良いとされている[20]．ここでは，Bidirectional LSTM とし，LSTM を両方向の形で利用している．その後，各時刻における Bidirectional LSTM の全出力（系列）を max pooling している．系列の max pooling は，テキスト分類でよく用いられる手法であり[21]，系列中の局所的な特徴が大きな値を持つように学習できた際に，max pooling によって，その値を最終的な出力に反映することができる．API 系列を用いてその API 群からマルウェアを分類する本タスクは，単語群からテキストを分類するタスクに類似していると考え，max pooling を採用した．

非時系列のデータには，全結合層を複数持つ順伝搬型のネットワークを利用する．また，それぞれにおいて，活性化関数には ReLU（Rectified Linear Unit）[22]を利用し，過学習を抑制するための Dropout 層を利用する．

上述の時系列・非時系列データを Concatenation 層で結合し，最終的に Softmax 関数によって出力を得る．Softmax 関数の出力は，各クラスへの所属確率となっている．つまり，解析対象のマルウェアが既存のマルウェアファミリのいずれに近いかを確率として出力する．

3.6 期待される効果

提案手法は，上述の手法によって，課題であった複数解析環境におけるマルウェアの動的解析レポートを活用したマルウェア分類を実現する．また，これによって以下の効果が期待できる．

(効果1) 亜種マルウェアの解析効率化

解析対象が既知のマルウェアをベースに作成されたような亜種マルウェアであれば，本手法によって，両者を結び付けることができる．これにより，人手での詳細な解析の省略や解析の効率化が期待できる．

(効果2) 新種マルウェアの抽出

解析対象が新種のマルウェアであれば，本手法では，既知のどのマルウェアにも結び付かず，それによって新種ではないかということが推定できる．このように，本手法によって，より重点的に解析すべき新種マルウェアを抽出することが期待される．

4. 評価

4.1 データセット

今回の評価では，FFRI Dataset 2016[23]を利用した．同データセットは，マルウェア対策研究人材育成ワークショップ（MWS）において提供されている研究用データセットの 1 つであり，Cuckoo sandbox[24]を用いたマルウェア 8,243 検体の解析結果である．同データセットには，同一検体を Windows 10，Windows 8.1 それぞれで解析した結果が収録されていることから，多種解析環境での解析結果として利用できると考え，今回の評価に利用した．

また，訓練・評価する際の教師ラベルとしては，Cuckoo sandbox によって標準化された ESET-NOD32 の検知名を用いた．これは，検知漏れや特定の検知名への偏りが比較的少なかったためである．なお，データセットのうち，ESET-NOD32 が検知したものについてのみ利用しており，その総数は，6,513，ユニークなラベル数は，315 である．ESET-NOD32 の FFRI Dataset 2016 に対する検知結果を図 3 に示す．青い棒がファミリ毎の検知数，オレンジの線がデータセット全体に対する累積分布を表している．

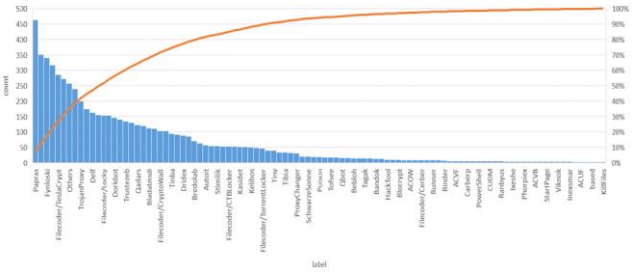


図 3 ESET-NOD32 によって検出されたマルウェアのファミリ毎の数とその累積分布

表 3 評価に用いたネットワークのパラメータ

層種	入力サイズ	出力サイズ	関数
Fully-Connected 1	25	25	ReLU
Fully-Connected 2	25	25	ReLU
Embedding	294*系列長	128*系列長	-
Bidirectional LSTM	128*系列長*2	128*系列長*2	-
Max pooling	128*系列長*2	128*2	-
Concatenation	25+256	281	-
Fully-Connected 3	281	134	Softmax

表 4 評価環境

CPU	Intel Core i7-6700
GPU	GeForce GTX 1080
メモリ	8,192 MB
OS	Ubuntu 16.04 LTS

4.2 評価項目

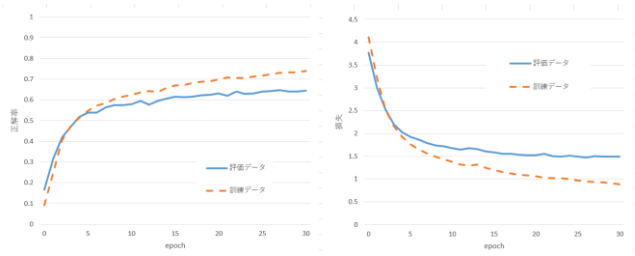
本稿では，3.6 節で述べた 2 つの効果がどの程度実現可能性を有しているか検証するために，以下に示す 2 つの評価を行う．

(評価1) 亜種マルウェアの分類精度

(評価2) 新種マルウェアの抽出可能性

今回利用するデータセットには，ユニークなラベル 315 のうち，2 検体以上の検体が含まれるものが 134，1 検体しか含まれないものが 181 ある．そこで，まずは前者の 134 を用いて分類精度を評価する．その後，1 検体しかいないものを擬似的な新種マルウェアとして新種マルウェアの抽出可能性を評価する．

評価に用いたネットワークのパラメータを表 3 に示す．今回用いたデータセットには，294 種類の API が含まれていたことから，Embedding 層への入力は 294 次元に，分類先の既知マルウェアは 134 種類であることから出力は 134 次元になっている．このとき，optimizer に AdaGrad[25]，Dropout 層におけるドロップアウト率に 50%，エポック数に 50 を用いている．ミニバッチサイズには，32 を利用し，可変長である API 呼び出し列をバッチ処理するために，パディングを挿入し，長さを統一している．また，多クラス分類であるため，損失関数には，交差エントロピーを用いている．さらに，過学習抑制のために，early stopping を利用している．なお，評価環境は，表 4 に示すとおりである．



(A) 正解率　　(B) 損失

図 4 学習時の値の推移

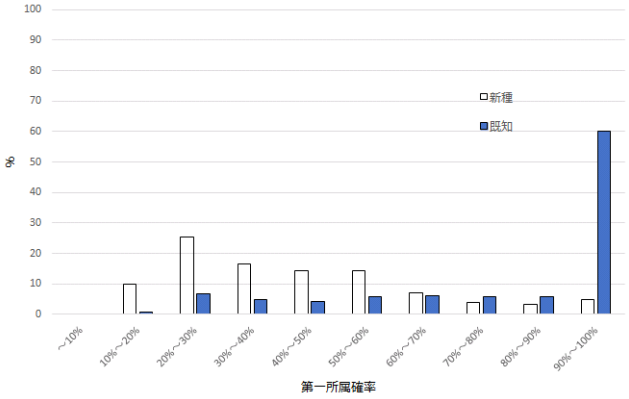


図 5 所属確率が最も高いファミリへの
所属確率のヒストグラム

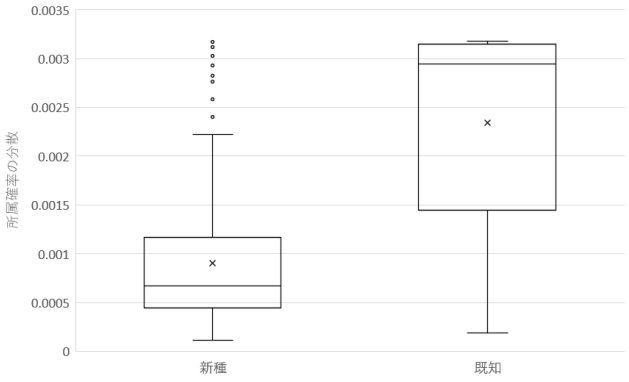


図 6 所属確率の分散の箱ひげ図

4.3 亜種マルウェアの分類精度

本評価では．ESET-NOD32 において同じラベルが付けられた検体を同じクラスに分類できるか検証した．データセット内に 2 検体以上含まれる 134 種類の検体のうち，ランダムに 8 割を訓練データ，2 割を評価データとした．前述のとおり，出力は各クラスへの所属確率となることから，出力のうち，最も所属確率の高いクラスと正解クラスが一致した場合に，分類に成功したとして，評価データを分類した際の正解率を算出する．

エポック毎の(A)正解率と(B)損失の推移を図 4 に示す．それぞれにおいて，訓練データに対する値をオレンジ，評価データに対する値を青で示している．まず，early stopping によって，エポック数 31 の時点で学習が打ち切られている．また，評価データに対する性能は，いずれもエポック数 28 の時点で，正解率が約 64.80%，損失が約 1.47 となっている．正解率約 64.80%という値は，既存研究に比べると改善の余地がある値である．例えば，文献[16]では，20 ファミリの 6000 検体を複数のアルゴリズムを用いて分類しており，いずれのアルゴリズムにおいても約 85%以上の正解率で分類している．データセットやファミリ数が異なるため，単純な比較はできないものの，正解率に劣る結果となっている．この結果に関する考察については，5 章で後述する．

4.4 新種マルウェアの抽出可能性

本評価では，提案手法において，訓練データに含まれない未知の検体を新種のマルウェアであるとして検出できるか検証した．ここで，新種マルウェアは，既知マルウェアのいずれにも強く分類されないことから，各ファミリへの所属確率やその分散が既知マルウェアに比べて小さいのではないかという仮説を立て，所属確率の違いによって新種マルウェアを抽出できないか実験を行った．本仮説を検証すべく，(評価 1)で構築したモデルに，既知マルウェアのデータセットとして同じく(評価 1)で用いた評価用のデータを，新種マルウェアのデータセットとして，FFRI Dataset 2016 に 1 検体だけ含まれる 181 種類の検体を投入し，各ファミリへの所属確率を出力した．なお，同評価を実施する上でのノイズを抑制するため，既知マルウェアのデータセットとして用いた評価用データには，(評価 1)において正しく分類できたもののみを利用した．

上述の実験によって出力された値のうち，最も所属確率の高いファミリへの所属確率（以降，第一所属確率）を 10%単位で区切り，新種マルウェア・既知マルウェアそれぞれをプロットしたヒストグラムを図 5 に示す．また，所属確率の分散を取り，箱ひげ図にしたものを図 6 に示す．

図 5 から分かるように，既知マルウェアの第一所属確率は，90～100%の範囲が最も高く，60～70%，70～80%の順に続く（それぞれ全体の 60.19%，6.04%，5.92%）．これに対し，新種マルウェアは 20～30%の範囲が最も高く，30～40%, 40～50%の順に続く（それぞれ全体の 25.41%, 16.57%, 14.36%）．このように，新種マルウェアは比較的第一所属確率が低いのに対し，既知マルウェアの第一所属確率は比較的高いことが確認できた．また，図 6 に示した所属確率の分散は，第一四分位数，中央値，および第三四分位数が既知マルウェアでは 0.00044，0.00067，および 0.0012，新種マルウェアでは 0.0014，0.0029，および 0.0031 となっている．このように，両者の新種マルウェアは所属確率の分散が比較的小さいのに対し，既知マルウェアは所属確率の分散が比較的大きいことが分かる．これは，仮説の通り，新種マルウェアは，既知マルウェアのいずれにも強く分類されずに一様な値をとり，分散が小さくなるのに対し，既知マルウェアは，自身のファミリへの所属確率は大きく，それ以外への所属確率は小さくなることからその分散が大きくなっていることが原因だと考えられる．

以上の実験から，第一所属確率や所属確率の分散を利用することによって，新種マルウェアが抽出でき得ることが確認できた．一方で，第一所属確率，所属確率の分散いずれに関しても，定性的な評価に留まっていることから，定量的な新種マルウェア抽出アルゴリズムを検討する必要がある．また，本手法は，既知マルウェアの分類性能に強く依存するため．同分類性能を洗練し，分類正解率を向上していく必要もある．

5. 考察

5.1 解析環境

解析妨害機能の 1 つに，マルウェア解析が仮想環境で行われることを利用し，仮想環境では動作を行わないものがある．このような解析妨害機能をもつ検体は，仮想環境と物理環境での動作の差異を見ることによって，検出することが可能である．一方で，今回の評価に用いたデータセットの解析環境は，OS は異なる（Windows 10，Windows 8.1）ものの，ともに仮想環境である．このため，上述のような解析妨害機能をもつ検体は，異なる動作を取らず，今回特徴量の 1 つとして用いた解析環境間での違いがあまり分類に寄与してないと考えられる．FFRI Dataset を用いた今回の実験によって，提案手法の利用可能性を検証できたため，今後は，M3AS などを利用し，仮想・物理環境の混在を含む，より複数の環境における環境での解析結果を用いて，より詳細な評価を実施する．

5.2 ラベル付与方法

今回は，実験に用いたデータのラベルとして単一のアンチウイルスソフトでの検知名を利用した．また，多くの先行研究においてもラベル付けに同様のアプローチがとられている．一方で，単一のアンチウイルスソフトでの検知名を用いた場合，ベンダのバイアスが混入しうることや検知名が ground truth と一致しない場合があることが他の先行研究によって指摘されている[26][27][28]．これらの問題を緩和し，ground truth なラベルを付与するべく，複数のアンチウイルスソフトでの検出結果を組み合わせる手法が用いられている[26][29][30][31]．厳密な訓練・評価のために，上述の手法を取り入れ，訓練・評価データに対してより高精度なラベル付けを行うことが望ましいと考えられる．

5.3 最大動作レポートの抽出

今回は，呼び出し API 数が最も多いものを最も動作した解析レポートとして抽出し，学習・分類に利用した．一方で，解析環境を検知すると無意味な API 呼び出しを繰り返すような解析環境妨害機能も存在する．このような場合，呼び出し API 数は多くなり，最も動作した解析レポートとして抽出されるものの，解析のために本来の挙動を最も発現させた解析レポートを抽出したいという意図には沿っておらず，適切な分類ができない恐れがある．このような現象の防止策として，例えば，マルウェアらしい挙動に利用される API には大きい重みを，そうではない API には小さい重みを割り振り，前者をより重用して呼び出し API 数をカウントすること等が考えられる．レポートの抽出方法については，引き続き検討していく．

5.4 ネットワーク選定

今回は，時系列データを扱うものとして，Bidirectional LSTM を用いたが，LSTM を多段に利用することによって，より中長期的な特徴を掴む Stacked LSTM や CNN（Convolutional Neural Network）を活用し，複数のタスクで LSTM より高精度を出している QRNN（Quasi-Recurrent Neural Networks）[32]もあり，これらを取り入れることによって正解率に改善が見られる可能性もある．また，非時系列側のネットワークに関しても，ネットワーク構造や各種パラメータを決めうちで構築した面があり，これらを洗練していくことでより正解率を向上できる可能性がある．

5.5 提案手法の運用

提案手法を運用する際，良性のプログラムは，既知マルウェアのいずれにも分類されず，新種マルウェア抽出の妨げとなる可能性がある．このため，2 章関連研究の(2)で述べたような良性プログラムとマルウェアを分別する手法と組み合わせ，マルウェアと判断されたもののみを提案手法に投入するといった運用が望ましいと考えられる．

6. おわりに

本稿では，マルウェア解析の自動化・効率化を目的に，従来手法における解析環境ではマルウェアが動作しない場合があるという課題を緩和するために．多種解析環境におけるマルウェアの動的解析結果を用いた分類手法を提案した．本手法は，解析対象のマルウェアを類似する既存のマルウェアファミリに分類するものであり，これにより，亜種マルウェアの選別や既存のマルウェアファミリに分類されない新種のマルウェア抽出に応用できること，およびそれらを通した解析全体の効率化が期待できる．また，プロトタイプを実装し，評価を行ったところ，既知のマルウェアを約 64.80%の正解率で分類できること，各マルウェアファミリへの所属確率を利用することによって，新種マルウェアを抽出でき得ることが確認できた．ただし，この結果は，既存手法と比較すると正解率で劣る点があり，改善の必要がある．

今後の課題としては，提案手法をより洗練することによる正解率の向上や新種マルウェアの抽出アルゴリズムの検討・洗練がある．また，本手法が解析環境検知型のマルウェアに対してどの程度頑強かを評価することが挙げられる．さらに，今後は，我々の研究グループで研究開発している多種解析環境 M3AS でマルウェアの解析を行い，そこから得られたデータで評価することを検討している．

参考文献

[1] theguardian: Antivirus software is dead, says security expert at Symantec, available from <https://www.theguardian.com/technology/2014/may/06/antivirus-software-fails-catch-attacks-security-expert-symantec> (accessed. 2017-08-10).

[2] Anubis: Malware analysis for unknown binaries, available from <https://anubis.iseclab.org/> (accessed 2016-02-20).

[3] A., Dinaburg, P., Royal, M., Sharif, and W., Lee: Ether: Malware analysis via hardware virtualization extensions, in Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS '08), pp. 51–62 (2008).

[4] C., Willems, T., Holz, and F., Freiling: Toward automated dynamic malware analysis using CWSandbox, Security Privacy, IEEE, Vol. 5, no. 2, pp. 32–39 (2007).

[5] U., Bayer, C., Kruegel, and E., Kirda: TTAnalyze: A tool for analyzing malware, in Proceedings of the 15th European Institute for Computer Antivirus Research Annual Conference (EICAR), (2006).

[6] H., Yin, D., Song, M., Egele, C., Kruegel, and E., Kirda, Panorama: Capturing system-wide information flow for malware detection and analysis, in Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07), pp. 116–127 (2007).

[7] Dhilung, K., Giovanni, V., Christopher, K.: BareCloud: Bare-metal Analysis-based Evasive Malware Detection, 23rd USENIX Security Symposium (USENIX Security 14), 287－301 (2014).

[8] 仲小路博史，重本倫宏，鬼頭哲郎，林 直樹，寺田真敏，菊池浩明：多種環境マルウェア動的解析システムの提案および評価，情報処理学会論文誌，Vol.56，No.9，pp.1730-1744 (2015).

[9] Shafiq, M., Z., Tabish, S., M., Mirza, F., and Farooq, M.: PE-Miner: Mining Structural Information to Detect Malicious Executables in Realtime, Proceedings of the 12th International Symposium on Recent Advances in Intrusion Detection (RAID '09), pp.121－141 (2009).

[10] Kolter, J., Z., and Maloof, M., A.: Learning to Detect Malicious Executables in the Wild, Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04), pp.470－478 (2004).

[11] Ahmed, F., Hameed, H., Shafiq, M., Z., and Farooq, M.: Using Spatio-temporal Information in API Calls with Machine Learning Algorithms for Malware Detection, Proceedings of the 2Nd ACM Workshop on Security and Artificial Intelligence (AISec '09), pp. 55－62 (2009).

[12] Rohitab.com: API Monitor, available from <http://www.rohitab.com/apimonitor > (accessed 2017-08-10).

[13] R., Tian, R., Islam, L., Batten, and S., Versteeg: Differentiating malware from cleanware using behavioral analysis, 2010 5th International Conference on Malicious and Unwanted Software, pp.23－30 (2010).

[14] Iwamoto, K., and Wasaki, K.: Malware Classification Based on Extracted API Sequences Using Static Analysis, Proceedings of the Asian Internet Engineeering Conference (AINTEC '12), pp.31－38 (2012).

[15] Christodorescu, M., Jha, S., and Kruegel, C.: Mining Specifications of Malicious Behavior, Proceedings of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering (ESEC-FSE '07), pp.5－14 (2007).

[16] Rafique, M., Z., and Chen, P., Huygens, C., and Joosen, W.: Evolutionary Algorithms for Classification of Malware Families Through Different Network Behaviors, Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation (GECCO '14), pp.1167－1174 (2014).

[17] Lindorfer, M., Kolbitsch, C., and Milani C., P.: Detecting Environment-sensitive Malware, Proceedings of the 14th International Conference on Recent Advances in Intrusion Detection, (RAID '11), pp.338－357 (2011).

[18] S., Honton, and J., Schmidhuber: Long Short-Term Memory, Neural Comput, No. 8, Vol. 9, pp.1735－1780 (1997).

[19] M., Schwartz, and K., K., Paliwal: Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing, pp. 2673－2681 (1997).

[20] A., Graves: Supervised sequence labeling with recurrent neural networks, PhD thesis, Technische Universitat Munchen (2008).

[21] Siwei, L., Liheng, X., Kang, L., and Jun, Z.: Recurrent Convolutional Neural Networks for Text Classification, Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI '15), pp.2267－2273 (2015).

[22] V., Nair and G., E., Hinton: Rectified linear units improve restricted Boltzmann machines, In Proceedings of the 27th International Conference on Machine Learning (ICML '10), pp. 807－814 (2010).

[23] 高田雄太，寺田真敏，村上純一，笠間貴弘，吉岡克成，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 2016 Datasets～，情報処理学会研究報告書，Vol.2016-CSEC-74， No.17，pp.1－8 (2016).

[24] Cuckoo Foundation : Automated Malware Analysis, available from <http://www.cuckoosandbox.org/> (accessed. 2017-08-10).

[25] John, D., Elad, H., and Yoram, S.: Adaptive Subgradient Methods for Online Learning and Stochastic Optimization, Journal of Machine Learning Research, Vol. 12, pp.2121－2159 (2011).

[26] Kantchelian, A., Tschantz, M., C., Afroz, S., Miller, B., Shankar, V., Bachwani, R., Joseph, A., D., and Tygar, J., D.: Better Malware Ground Truth: Techniques for Weighting Anti-Virus Vendor Labels, Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security (AISec '15), pp.45－56 (2015)

[27] Sheng, V., S., Provost, F., and Ipeirotis, P., G.: Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers, Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08), pp.614－622 (2008).

[28] Perdisci, R., and U., ManChon: VAMO: Towards a Fully Automated Malware Clustering Validity Analysis, Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC '12), pp.329－338 (2012).

[29] Perdisci, R., Lanzi, A., and Lee, W.: McBoost: Boosting Scalability in Malware Collection and Analysis Using Statistical Classification of Executables, Proceedings of the 2008 Annual Computer Security Applications Conference (ACSAC '08), pp. 301－310 (2008).

[30] Laskov, P., and Srndic, N.: Static Detection of Malicious JavaScript-bearing PDF Documents, Proceedings of the 27th Annual Computer Security Applications Conference (ACSAC '11), pp.373－382 (2011).

[31] Gascon, H. and Yamaguchi, F., Arp, D., and Rieck, K.: Structural Detection of Android Malware Using Embedded Call Graphs, Proceedings of the 2013 ACM Workshop on Artificial Intelligence and Security, (AISec '13), pp.45－54 (2013).

[32] James, B., Stephen, M., Caiming, X., and Richard, S.: quasi-recurrent neural networks, 5th International Conference on Learning Representations (ICLR '17) (2017).