6X-05

機械学習を用いたマルウェア検出手法の検討

厚地紗江香† 平川 豊‡
†芝浦工業大学大学院理工学研究科 ‡芝浦工業大学工学部情報工学科

1. 研究背景

近年、社会の情報化が推し進められ、様々な情報がインターネットを介して共有される機会が多くなり、サイバー攻撃、特にそれに用いられるマルウェア（悪意のあるソフトウェア）の脅威と被害は年々増加している[1]。また同時に、未知のマルウェアや既存のマルウェアの亜種が増加していることから、従来のシグネチャ（攻撃を識別するルール）とのパターンマッチング方式によるマルウェア検出では、シグネチャの作成とその検出モデルへの適用にタイムラグが生じるという課題がある。そのため、より効率的でスケーラブルなマルウェア検出手法が求められる。そこで、機械学習を用いたマルウェア検出手法が活発に研究されている。

機械学習には入力となる特徴量（説明変数）と出力となる予測値（目的変数）が必要であり、特に特徴量の選択はモデルの精度を左右する。ここでは、API（アプリケーションインターフェース）呼び出し列からのマルウェア検出手法に着目する。先行研究では、API の順序関係に着目して特徴抽出を行っている研究は少ない。API の順序関係はマルウェアの処理プロセスを表現し、挙動を定義する上では重要なポイントである。

本研究では、API 呼び出し列の順序関係に着目した特徴量抽出を行い、その特徴量を用いた機械学習モデルを検討する。

２．先行研究

先行研究[2]では深層学習の代表的な手法である Recurrent Neural Network(RNN)を用いた複数のマルウェア検出手法を比較検討している。データに対して時系的に更新される隠れ層の出力（以降これを状態ベクトルと呼ぶ）を利用している。その１つに、API 呼び出し列の中間時点の隠れ層と最終時点の隠れ層を結合したものをマルウェアの特徴量とする手法（以降二分割手法と呼ぶ）を提案しており、抽出後はロジスティック回帰と多層パーセプトロンを用いて推論モデルを構築している。

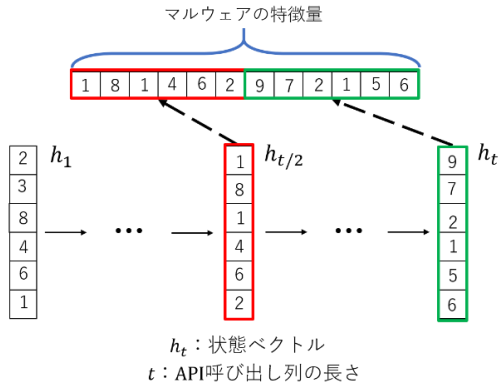


図 1. 二分割手法の概略図

Malware Detection Methods Using Machine Learning
†Saeka Atsuchi, ‡Yutaka Hirakawa
†Electrical Engineering and Computer Science, Shibaura Institute of Technology, Tokyo, Japan
‡Computer Science and Engineering, Shibaura Institute of Technology, Tokyo, Japan

2. 提案手法

2.1 手法概要

本研究で提案する手法は、特徴抽出段階と推論段階で構成されている。

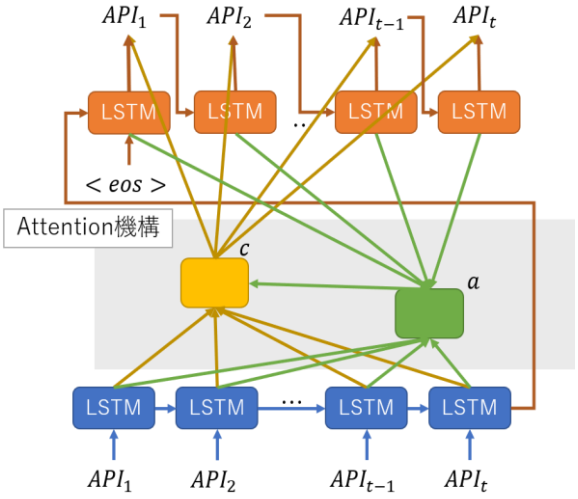
2.2 特徴抽出段階

特徴を抽出する対象はマルウェアの API 呼び出し列である。

先行研究[2]では時系列データの学習に特化した RNN が利用されたが、これでは時系列データの長期的な依存関係を出力に反映できない。よって API 呼び出し列が長い場合、正確な特徴量を抽出することが出来ない。

そこで RNN の中でも、入力列の長期的な依存関係を記憶することのできる Long Short Term Memory(LSTM)モデルと入力列の情報から、どの部分が重要であるか知らせる Attention 機構を組み合わせたモデル(以降 LSTM+Attention 機構モデル)、さらに、適切な分割数を選択した上で、分割手法を採用することで、効率よく API 呼び出し列の順序関係を特徴量に反映させることを目指す。

具体的には以下の図のような LSTM+Attention 機構モデルを用いている。まず、LSTM に API 呼び出し列をそれぞれ入力し、各時刻の隠れ層と Attention 機構から重要度の情報を得る。そして、分割点にあたる隠れ層と重要度から、その時刻の状態ベクトルを算出する。



c：各時刻の隠れ層の出力の加重平均
a：重み(重要度)

図 2.LSTM+Attention 機構モデル

特徴抽出手順の概要を以下に示す。

① マルウェアの API 呼び出し列を収集(本研究では既存のデータセットを使用)
② 各 API を one-hot ベクトルに変換
③ 入力を時刻 t の API の one-hot ベクトル、出力を時刻 t+1 の API の one-hot ベクトルとして LSTM+Attention 機構モデルを学習させる
④ 学習済みの LSTM+Attention 機構モデルに調査対象となるマルウェアの API 呼び出し列を③と同様に入力する
⑤ 設定した分割点に応じた状態ベクトルを順に結合したものを特徴量とする

2.3 推論段階

先行研究[2]では抽出した特徴量で、マルウェアであるか、グッドウェアであるかを学習し、それぞれの推論モデルを構築する。推論モデルは未知のソフトウェアの API 呼び出し列に対して、推論を行い、グッドウェアを 0 またはマルウェアを 1 として判定する。学習には多層パーセプトロンとロジスティクス回帰を用いた場合を比較検討している。

本研究では、前述した 2 つに加えて、サポートベクタマシン、k 近傍法、ランダムフォレストを用いて検出モデルを構築し、比較評価を行う。

3. 評価・実験

3.1 使用するデータセット

本実験では Angelo[3]によって公開、共有されているデータセットを使用する。このデータセットには 42797 個のマルウェア API 呼び出し列と 1079 個のグッドウェア API 呼び出し列が含まれている。また、マルウェアとグッドウェアのサンプル数に偏りがあるため、マルウェアの個数をグッドウェアの個数に合わせる処理を行っている。

3.2 予備実験

分割手法における最適な分割数を明らかにするため、先行研究[2]の二分割手法と同様のアルゴリズムで、分割数のみを変化させた特徴抽出を行い、それによる予測精度の変化を調査する。データセットは 3.1 で記述したものを用いる。

結果はテストデータに対するものであり、以下の表の通りである。評価指標は 1 に近いほど判別能が高いことを表す、ROC(Receiver Operating Characteristic)曲線の Area Under the Curve(AUC)を用いている。N は分割数を表しており、N=2 は先行研究[2]の二分割手法である。また、ロジスティック回帰を LR、多層パーセプトロンを MLP で表している。

表 1.分割数(N)による、AUC の変化

	N=2	N=3	N=4	N=5
LR	0.8688	0.8857	0.9155	0.9058
MLP	0.9167	0.9274	0.9481	0.9374

結果は分割数を変化させることで、予測精度が向上した。そのため、分割数については詳細な調査をする必要があることがわかった。

3.3 実験

提案手法の有効性を確認するため、予備実験と同様に、3.1 のデータセットを用いて、マルウェアか、グッドウェアであるかを推論する実験を行う。

特徴抽出モデルの学習には、全てのデータを使用する。LSTM のそれぞれの次元数は、入力層はデータセットに含まれる API の種類数により 307 次元、隠れ層の次元数は 100 次元、出力層は 307 次元である。最適化アルゴリズムは Adam、損失関数は交差エントロピー誤差、エポック数は 30 である。使用するモデルはエポックごとに損失の平均が一番少ないモデルを使用する。

推論モデルの学習では、データセットの 70%を訓練データとし、残りの 30%をテストデータとして利用する。先行研究[2]で採用されたロジスティック回帰と多層パーセプトロンに加えて、サポートベクタマシン、k 近傍法、ランダムフォレストの合計 5 つの推論モデルを構築する。

多層パーセプトロンは 2 つの隠れ層を持ち、どちらとも 1024 次元、この活性化関数は Rectified Linear Unit である 。サポートベクタマシンのカーネルには rbf カーネル、k 近傍法の最近傍個数(k)は 3、サンプルと近傍の距離の計算式はミンコスフキー距離を採用する。ランダムフォレストの max_depth は 4 である。以上の条件で実験を行なった。

3.4 評価

評価環境は以下の通りである。

4. 表 2.評価環境

CPU	Intel Core i7-9700F CPU @ 3.00GHz
GPU	GeForce RTX 2070 SUPER
メモリ	16GB
OS	Ubuntu 20.04.1 LTS
実装言語	Python3.8.5

提案する LSTM+Attention 機構と適切な分割数による分割手法の有効性を確認するために、先行研究[2]、手法 1：LSTM+Attentio 機構モデル+分割手法(N=2)、手法 2：LSTM+Attention 機構モデル+分割手法(N=4)の比較評価を行う。分割数は先行研究で使用している N=2 と予備実験で最も値が良かった N=4 を評価対象とした。

評価指標は予備実験と同様に AUC であり、訓練データとテストデータに対する実験結果をそれぞれ表に示す。

5. 表 3.訓練データに対する実験結果

	先行研究[2]	手法 1	手法 2
LR	0.9059	0.9344	0.9522
MLP	0.9953	0.9960	0.9980
SVM	-	0.9514	0.9679
KNN	-	0.9389	0.9532
RF	-	0.9036	0.9263

6. 表 4.テストデータに対する実験結果

	先行研究[2]	手法 1	手法 2
LR	0.8688	0.8703	0.8973
MLP	0.9167	0.9626	0.9720
SVM	-	0.9351	0.9574
KNN	-	0.9304	0.9290
RF	-	0.8973	0.9137

7. 考察

最も AUC の値が高かったものは、訓練データ、テストデータともに手法 2+MLP であった。テストデータにおいては先行研究[2]よりも 0.03 以上高い 0.9720 の推論精度を得ることができた。これは、最適な分割点を選択し、かつ API 同士の依存関係や順序関係を特徴量に十分に反映できたためであると考える。

5.まとめと今後の課題

本研究では、機械学習を用いたマルウェア検出手法の検討を行った。予備実験により、先行研究の特徴抽出手法の改良の余地を見出し、新たなモデルと組み合わせ、多様な機械学習モデルに 2 値分類の実験を行なったことで、先行研究よりも良い推論精度を得ることができた。

今後の課題として、今回用いたデータセットはグットウェアのサンプル数が少ないものであったため、これを増やした際の詳細な検討が残される。

参考文献

[1] AV-TEST. Malware Statistics Trends Report. 2019.," https://www.av-test.org/en/statistics/malware/"

[2] Pascanu, R., Stokes, J.W., Sanossian, H., Marinescu, M., Thomas, A.: "Malware classification with recurrent networks.", Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP),pp.1916-1920. ,2015

[3] Schranko de Oliveira, Angelo; Sassi, Renato José ," Behavioral Malware Detection Using Deep Graph Convolutional Neural Networks.",TechRxiv. Preprint, "https://doi.org/10.36227/techrxiv.10043099.v1 ",2019