

深層学習によるディスクアクセスパターンを用いた ランサムウェア検知システム

程田 凌羽^{1,*} 平野 学^{2,*} 小林 良太郎^{3,*}

概要：一般的なランサムウェア検知システムではシグネチャ方式が採用されることが多く、それらのシステムではプログラムのハッシュ値やバイナリを静的解析して得られた特徴量を用いて検知をおこなっている。しかし、近年はランサムウェアの亜種が増加しており、それらの亜種に対応するシグネチャをすべて作成していくことが困難になりつつある。そこで本研究ではランサムウェアのストレージアクセスパターンを特徴量とした新しい振る舞い型の検知手法を検討する。先行研究では6種類のランサムウェア検体と3種類の無害なプログラムのストレージアクセスパターンを収集し、古典的な機械学習アルゴリズムを用いてモデルの訓練を行い、ランサムウェアの検知性能を評価した。本研究では同一のデータセットを用いて、新たに深層学習のモデルを訓練してランサムウェアの検知性能を評価した結果を報告する。

キーワード：深層学習, ランサムウェア, 振る舞い型検知

A Deep-learning-based Ransomware Detection System Using Disk Access Patterns

Ryo Hodota^{1,*} Manabu Hirano^{2,*} Ryotaro Kobayashi^{3,*}

Abstract: Many ransomware detection systems employ a signature-based method using features obtained from static analysis. However, as the number of ransomware variants has increased, creating signatures for all of these variants has become impractical. In this study, we have investigated a behavioral-based ransomware detection method using storage access patterns. In our previous study, we collected storage access patterns of six ransomware samples and three benign samples. We then trained machine-learning models using classic machine-learning algorithms and evaluated the performance of ransomware detection. In this paper, we present the results of the ransomware detection performance using deep learning.

Keywords: Deep learning, Ransomware, Behavioral-based detection

1. はじめに

近年、企業、官公庁、教育機関、病院等の公共サービス分野でのランサムウェアの被害が増加している。ランサムウェアとは、攻撃対象のストレージ装置に保存してあるファイルを暗号化し、復号鍵と引き換えに身代金を要求するマルウェアである。最近ではデータを公開する二重恐喝型のランサムウェアも増加している。ビットコインなどで身代金を支払ったとしてもデータが回復される保証はなく、攻撃者を特定しても公開されたデータを回収できるとは限らない。そのため、ランサムウェアに感染しないことが重要であり、様々なタイプの検知システムが提案されている。

ランサムウェアをはじめとするマルウェアの脅威を検知する手法には2つのアプローチが存在する。ひとつは Misuse detection (悪用検知)、もうひとつは Anomaly detection (異常検知) である。前者は過去の悪用をモデル化して検知するのに対し、後者は定常状態をモデル化してそ

れ以外を検知する。後者の Anomaly detection はゼロデイといわれる未知の攻撃を検知できる点で Misuse detection に勝っている。しかし、Anomaly detection には多様なユーザ環境をモデル化することは極めて難しいという問題を抱えている。このため実際のアンチウイルス製品は基本的に Misuse detection での設計がなされており、その中でも特にシグネチャ型による検知が採用されていることが多い。

シグネチャ型の検知は、検査対象をシグネチャと比較することで検知する。ここでシグネチャとは、特定のマルウェア検体の特徴量を示すデータであり、ファイルのハッシュ値やバイナリを静的解析することで得られる API コールグラフ等がある。しかし、シグネチャ型の検知を回避するために、パッカーを用いた難読化や暗号化を施した亜種も作られており、最近では深層学習と静的解析を組み合わせたシグネチャ型検知システムを回避する亜種を自動的に作成する研究も行われている[1]。シグネチャの作成は専門の技術者が手作業で解析して行うため、検体の数に比例してシ

¹ 豊田工業高等専門学校 専攻科
National Institute of Technology, Toyota College, Advanced Course
² 豊田工業高等専門学校
National Institute of Technology, Toyota College.

³ 工学院大学
Kogakuin University

グネチャを作成するコストも増大する。シグネチャ型にはこれらの欠点があるものの、誤検知が少ないという利点があるため実用化に適した手法として定着している。

最近ではシグネチャ型の欠点を補い、新種や亜種の増加速度に対応するために振る舞い型で検知する研究も行われている。振る舞い型ではプログラムを実行したときに観測される特徴で判定する。Berrueta らはランサムウェアを検知するシステムを局所静的、局所動的、ネットワークの3つに分類した[2]。局所静的はプログラムのバイナリデータに含まれる文字列、実行コード、ファイルのハッシュ値などをシグネチャとして利用し、ランサムウェアが実行される前に検知できる利点がある。局所動的ではランサムウェアのふるまいを見て検知を行う。具体的には、ランサムウェアによって暗号化されたファイルのエントロピーの増加、追加されたファイルの拡張子、ファイルシステム関連のシステムコール呼び出しの頻度、アクセスされたファイル数とディレクトリ数、カナリア (canary) ファイルへのアクセスや変更操作、とった情報を入力としてランサムウェアを検知する。ネットワークでは DNS クエリの特徴、ドメイン生成アルゴリズムが生成したドメイン名の特徴、ファイルサーバへの通信の頻度などを利用する。

本研究は Berrueta らが示した 3 つのランサムウェア検知のカテゴリのうち局所動的に該当する。しかし、これまでの局所動的の研究の多くがオペレーティングシステムから得られる情報、例えば作成されたファイルの拡張子やファイルごとのエントロピー、ファイル操作に関連するシステムコール呼び出しの頻度を入力としていたのに対して、本研究ではデバイスドライバのレベルで得られるストレージ装置のアクセスパターン（セクタ単位のエントロピーや読み書きされたセクタ番号の分散）を利用する。

先行研究[3]では古典的な機械学習アルゴリズムである Random Forest を用いてランサムウェア 6 種類と無害な良性プログラム 3 種類のストレージアクセスパターンを学習させて評価を行った。本稿では先行研究[3]と同じデータセットを用い、新たに深層学習によるランサムウェア検知システムを訓練し、その性能を評価した結果を報告する。

2. ストレージアクセスパターンの収集

まず本稿で扱うストレージアクセスパターンとは何かを説明する。我々の先行研究[4]ではハイパーバイザを用いて、ゲスト OS に手を加えずにストレージ装置へのアクセス履歴を時系列で保存するシステムを提案した。本研究で利用した軽量ハイパーバイザ BitVisor [5] は仮想化層で入出力へのセキュリティ強制や監視（たとえば暗号化やアクセス制御など）をおこなうために設計されており、1 台の物理マシンで 1 つのゲスト OS だけを仮想化する。

特に、BitVisor は準パススルー方式を採用しており、セキュリティを強制させる必要のない入出力をそのままハード

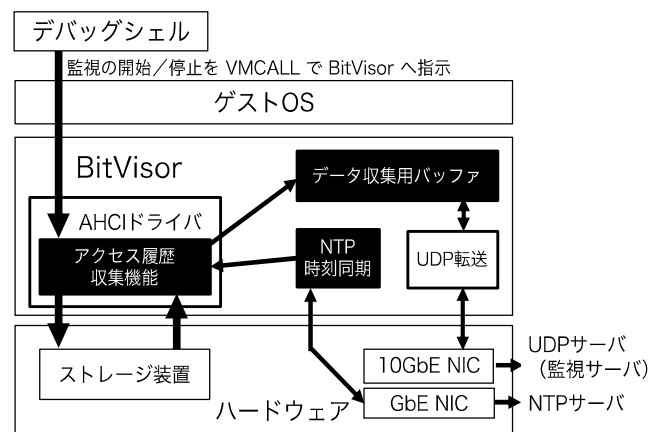


図 1 ストレージ装置へのアクセスパターンの収集

ウェアへ通過させることで性能の低下を最小限に抑えている。これはランサムウェアの振る舞いであるアクセスパターンをより正確に収集するのに適した設計といえる。

図 1 に本稿で用いたアクセスパターン収集システムを示す。このシステムでは BitVisor が提供する準パススルードライバのひとつ、Advanced Host Controller Interface (AHCI) ドライバを改造して Serial Advanced Technology Attachment (SATA) 方式のストレージ装置へのアクセスを収集している。収集したアクセスパターンは、BitVisor が提供する UDP スタックと 10 GbE ドライバを利用して監視サーバへ転送する。また、BitVisor は TCP/IP スタックも提供するため、これを用いて NTP クライアントを BitVisor に実装し、それぞれのアクセス履歴にタイムスタンプを記録できるようにした。収集するアクセスパターンのデータは以下の通りである。

- UNIX タイムスタンプ（秒，ナノ秒）
- Logical Block Address (LBA)
- アクセスの種類（読み込み，書き込み）
- 書き込みの場合，実際に書き込まれたデータ

Logical Block Address は論理セクタを参照するための番号である。本稿の実験では 1 セクタが 512 バイトのストレージ装置を利用した。本研究ではこのシステムを安全な隔離環境に配備し、ランサムウェアを実際に動作させてアクセスパターンを収集した。

3. 方法

3.1 深層学習をなぜ採用する必要があるのか

先行研究[3][6]では上述のストレージアクセスパターンを使用して Random Forest を用いた機械学習を行い、限られたサンプルを用いた実験ではあるものの、ランサムウェアを 99% 程度の精度 (F スコア) で検知できることを示した。本研究では同じストレージアクセスパターンのデータ

セットを使用して、新たに深層学習でランサムウェアの検知性能を評価する。機械学習で上記のような検知性能を達成しているにも関わらず、深層学習を使用する理由は以下のとおりである：（１）近年の深層学習が可能なハードウェアの普及、機械学習の推論は CPU で行われるが、深層学習は専用ハードウェアで行われることが多くなっている。たとえば最新の CPU には深層学習を処理する命令が提供され始めているほか、手頃な価格で GPU (Graphic Processing Unit) も入手できるようになっている。（２）データの预处理の自動化。先行研究[3][6]では収集したアクセスパターンを 10 s の時間ウィンドウをシフトさせて特徴量を生成する、いわゆる特徴エンジニアリングを施してから機械学習アルゴリズムに読み込ませている。我々は現在ストレージアクセスパターンに加えて、メモリのアクセスパターンも特徴量として追加する実装を進めており[7]、これらの特徴エンジニアリングを手作業でおこなうのではなく、Stacked Sparse Auto Encoder (SSAE) [8] や One-dimensional Convolutional Neural Network (1D CNN) [9]で自動抽出できないかと考えている。

3.2 本稿で検証すること

本研究で扱うデータは Intrusion Detection System (IDS) 用データセットの深層学習で扱っているようなシンボルのシーケンスを扱っているわけではなく、より低レベルなアクセスパターン、具体的にはアクセスされたストレージ装置のセクタの番地、そのセクタに書き込まれたデータ、を扱っている。IDS で利用された深層学習モデル（例えば Khan らによる Conv-LSTM [10]）をそのまま利用できないため、本稿では深層学習の基本的なモデルを検討することから始めることにした。

本研究では IDS の深層学習で利用されることの多い Long-short Term Memory (LSTM) [11] を検証する。LSTM は過去に入力されたデータを再帰的なバックプロパゲーションによって学習するニューラルネットワークであり、自然言語処理や音声認識、セキュリティ分野では IDS データセットから攻撃を検知するのに利用されている。本稿では LSTM だけで構成されるシンプルなニューラルネットワークを比較することで、最適なモデルとパラメータを調査した結果を報告する。

3.3 LSTM へ入力する訓練データ

本稿で LSTM に入力する訓練データは、先行研究[3][6]で用いたのと同じ 6 種類のランサムウェアと 3 種類の良性プログラムのストレージアクセスパターンから作成した。訓練データは、以下の 5 次元の特徴量を 1 秒間ごとの平均値として計算し、30 個 (30 秒分) のタイムステップ分並べたデータであり、それらに検体名のラベルをつけた。

表 1. ランサムウェア検体と良性プログラム

クラスラベル	ハッシュ値 (またはバージョン)
WannaCry	ed01ebfbc9eb5bbea545af4d01bf5f10 71661840480439c6e5babe8e080e41aa
TeslaCrypt	9b462800f1bef019d7ec00098682d3ea 7fc60e6721555f616399228e4e3ad122
Cerber	e67834d1e8b38ec5864cfa101b140aca ba8f1900a6e269e6a94c90fcbfe56678
Sodinokibi (REvil)	0fa207940ea53e2b54a2b769d8ab033a 6b2c5e08c78bf4d7dade79849960b54d
GandCrab4	c9941b3fd655d04763721f266185454b ef94461359642eecc724d0cf3f198c988
Ryuk	23f8aa94ffb3c08a62735fe7fee5799 880a8f322ce1d55ec49a13a3f85312db2
Zip	Windows7 標準
AESCrypt	3.10
SDelete	2.02

- 書き込まれたデータ量
- 書き込まれた LBA の分散
- 書き込まれたデータのエン트로ピー
- 読み込まれたデータ量
- 読み込まれた LBA の分散

検体の一覧を表 1 に示す。ランサムウェアが 6 種類と無害なプログラム 3 種類の計 9 種類の検体を用意した。さらに 120GB の HDD と 120GB の SSD の 2 種類のストレージ装置でアクセスパターンを収集した。ランサムウェアを実行させる際には、デスクトップに GovDoc1 データセット [12] から抽出したおとりファイルを置き、以下の 3 パターンの条件でアクセスパターンを収集した：（１）おとりファイルを何も置かない場合、（２）さまざまなサイズのファイルを 9,872 個置いた場合 (合計 5.2 GiB)、（３）10MB から 100MB のファイルを 605 個置いた場合 (合計 12.5 GiB)。条件（２）のファイルは GovDoc1 データセットの最初の 10 個のディレクトリ (000 から 009) から抽出した。

ランサムウェアは 3 条件でデータを収集したため 18 クラスとなり、無害なソフトウェアはおとりファイルの影響がないため 3 クラスとなり、計 21 クラスの分類問題となった。本稿では 21 クラス分類の結果と、ランサムウェアか良性プログラムかを判別する 2 クラス分類の結果を示す。

本稿の実験では、検体、実行環境、記憶媒体の組み合わせに対して、それぞれ 30 秒のストレージアクセスパターンを 10 回分、合計 420 サンプルを用意した。これらを訓練データと検証データに 1 : 1 で分けて評価をおこなった。訓練データと検証データの分割はランダムに行うが、実験の再現性を担保するために今回は乱数シードを固定して実験をおこなった。

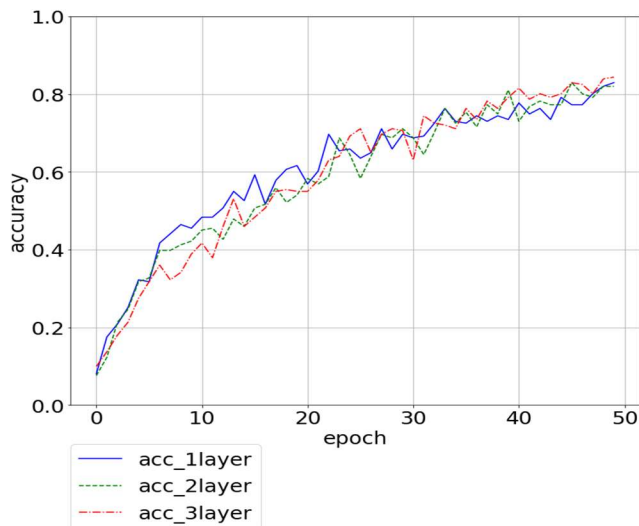


図 2 LSTM の層数と 21 クラス分類の正解率の変化
(トレーニング)

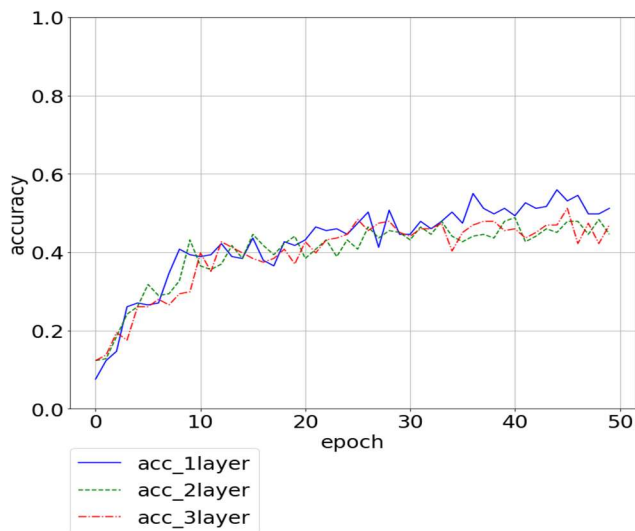


図 3 LSTM の層数と 21 クラス分類の正解率の変化
(バリデーション)

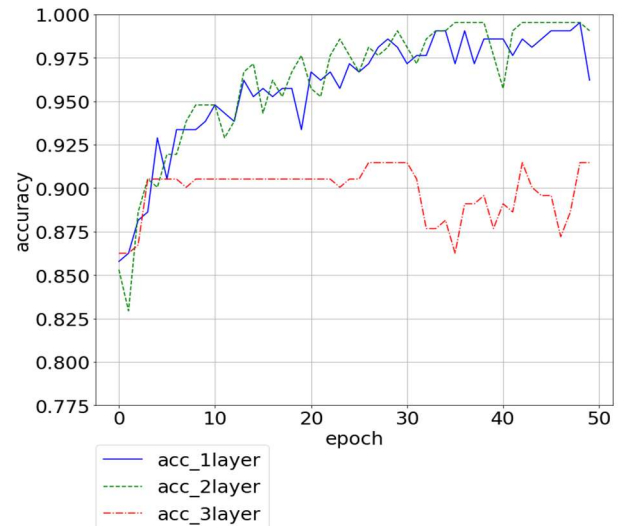


図 4 LSTM の層数と 2 クラス分類の正解率の変化
(トレーニング)

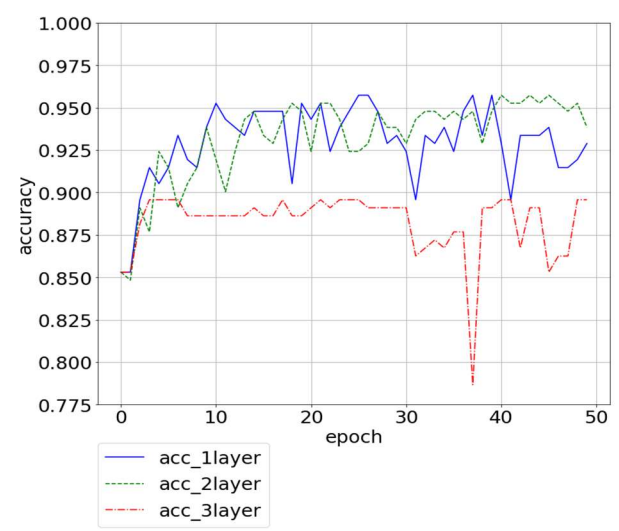


図 5 LSTM の層数と 2 クラス分類の正解率の変化
(バリデーション)

3.4 本稿で検証するハイパーパラメータ

本稿では LSTM の層数と、LSTM の隠れ状態の数が検知性能に与える影響を調査し、最適なパラメータを決定する。LSTM の層数については 1 層、2 層、3 層の場合の正解率を比較する。層数を変更する際には、すべての LSTM 層の隠れ状態を 32 に固定した。次に LSTM の隠れ状態の数の影響について、LSTM の隠れ状態を 1 から 128 まで 2 のべき乗で増やして正解率の変化を検証した。隠れ状態を変化させる際には LSTM は 2 層に固定した。ラベルは one-hot エンコーディングで与えた。損失関数は categorical cross-entropy を用いた。全条件において LSTM の後に全結合層を加えて 21 クラス分類または 2 クラス分類を行わせた。2 つの実験の両方について、先行研究[3]で得られた古典的な機械学習アルゴリズムである Random Forest の結果と比較する。

4. 結果

3 節で示した LSTM モデルと訓練データを用いて評価をおこなった。深層学習ライブラリには TensorFlow 2.1.0 と Keras2.3.1 を用いた。まず LSTM の層数を変えた場合の正解率の変化を調査した。図 2 に 21 クラス分類でのトレーニング時の正解率を、図 3 にバリデーション時の正解率を示す。図 4 に 2 クラス分類でのトレーニング時の正解率を、図 5 にバリデーション時の正解率を示す。次に、隠れ状態の数を変えた場合の正解率の変化を調査した。図 6 に 21 クラス分類でのトレーニング時の正解率を、図 7 にバリデーション時の正解率を示す。図 8 に 2 クラス分類でのトレーニング時の正解率を、図 9 にバリデーション時の正解率を示す。

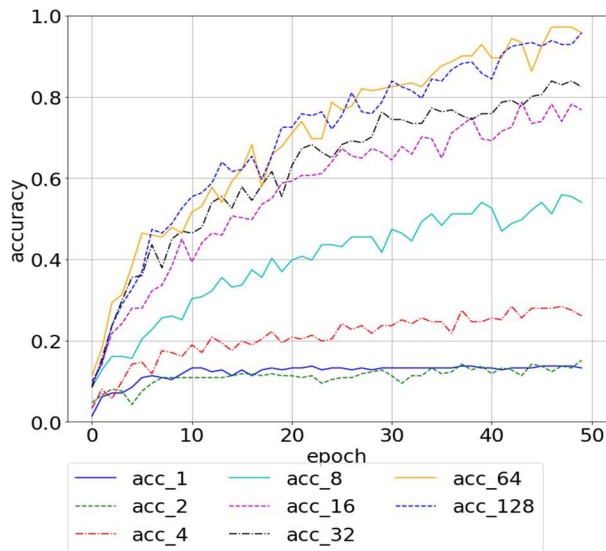


図 6 LSTM の隠れ状態の数と 21 クラス分類の正解率 (トレーニング)

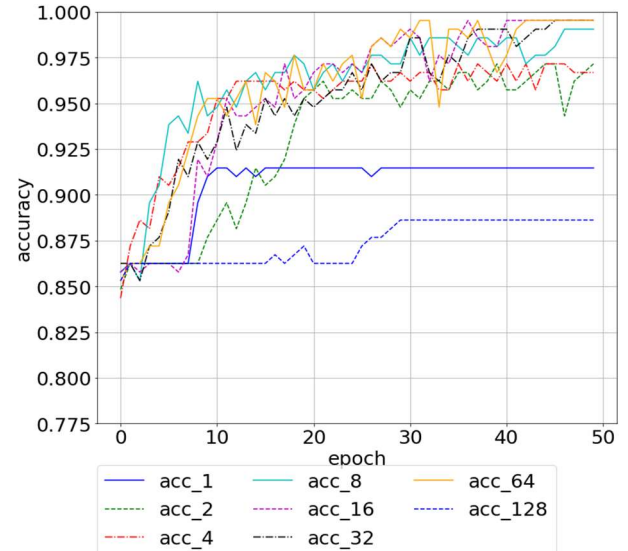


図 8 LSTM の隠れ状態の数と 2 クラス分類の正解率 (トレーニング)

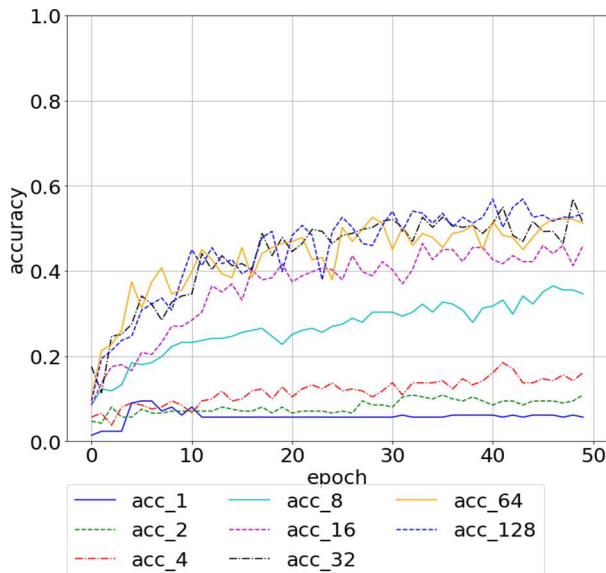


図 7 LSTM の隠れ状態の数と 21 クラス分類の正解率 (バリデーション)

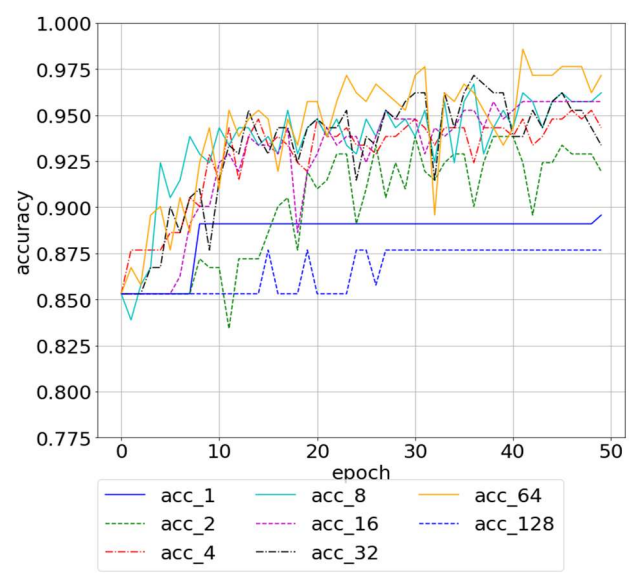


図 9 LSTM の隠れ状態の数と 2 クラス分類の正解率 (バリデーション)

5. 考察

5.1 最適なハイパーパラメータの検証

まず LSTM の層数による正解率の変化について考察する。図 2 の 21 クラス分類 (トレーニング) のグラフでは層数は正解率にあまり影響を与えなかったが、図 3 の 21 クラス分類 (バリデーション) のグラフでは、LSTM が 1 層の時に最も良い結果となった。一方、図 4 と図 5 の 2 クラス分類での正解率の変化を見ると、LSTM を 3 層にしたときに極端に正解率が低下した。また、図 4 と図 5 のどちらの結果においても LSTM 2 層のほうが LSTM 1 層よりも若干正解率が高く、安定した。今回の実験で用いた訓練データを用いた場合には、LSTM は 1 層または 2 層が適していることを確認できた。

次に隠れ状態の数が正解率に与える影響を考察する。図 6 と図 7 のグラフより、隠れ状態の数が増えるにつれて最終的な正解率も上昇した。図 6 の場合は隠れ状態が 32 以上で正解率が 0.8 程度まで到達し、図 7 では隠れ状態が 32 以上で正解率が 0.5 程度まで上昇した。一方、2 クラス分類問題である図 8 をみると隠れ状態が 128 の場合の正解率が極端に低く、隠れ状態が 1 の正解率よりも低くなってしまっている。また、図 8 では隠れ状態が 64, 32, 16 の時に正解率が最高であったのに対し、図 9 では隠れ状態が 64 の時に (ばらつきが大きいものの) 正解率が最も高くなった。以上の 2 つの実験結果をまとめると、今回の訓練データでは LSTM 層を 2 層、隠れ状態を 64 にするのが最適な組み合わせであると考えられる。

5.2 正解率とハイパーパラメータに関する考察

2 クラス分類において LSTM 層を 3 層にした場合や、隠れ状態を 128 にした場合に正解率が低下してしまった原因としては、学習に使用されるパラメータ数に対して分類すべきクラス数が少なくなり、各クラスの特徴が逆に上手く学習できなくなってしまうことが考えられる。

また、21 クラス分類時のトレーニングの正解率が高くなっているのに対し、バリデーションの正解率があまり上昇していない原因としては、訓練データの少なさが考えられる。21 クラス分類での訓練データは 1 クラスあたり 5 サンプル (30 秒のデータが 5 回分) しかなく、特徴がうまく学習できていないと考えられる。根本的な対策としてはデータをさらに収集して訓練データを増加させることであるが、ほかにデータ拡張によってデータ数を水増しすることも検討する。また、訓練データと検証データの分割に偏りが生じないように工夫をすることも必要である。

5.3 ランサムウェア検知への適用に向けて

本稿で示した深層学習を用いた結果のうちで最も高い正解率であった LSTM 2 層で隠れ状態が 64 のモデルでの正解率と、先行研究[3]の機械学習の正解率を比較した結果、21 クラス分類と 2 クラス分類のどちらにおいても、機械学習のほうが高い正解率であった。21 クラス分類の正解率の差は 10% 程度であり、今後はさらに深層学習の正解率を改善していく必要がある。実際のランサムウェア検知で必要となる 2 クラス分類においては、本稿で示したように深層学習でも 98% の正解率を得ることができ、機械学習に近い性能を有していることを確認できた。

先行研究[3][6]では特徴エンジニアリングを手作業でこない、30 秒のストレージアクセスパターンを 10 s ごとの時間ウィンドウで特徴量の平均値をもとめ、そのウィンドウをシフトさせていくことで 10 s ごとの時系列データを 5 つの特徴量にまとめる前処理をしていた。対して、本稿で示した深層学習では、1 秒間の平均値を特徴量としてまとめたものの、それらのデータを順番に LSTM に入力していくことでシーケンスを LSTM に学習させた。つまり時間ウィンドウをつかった特徴エンジニアリングを省略してもそれなりの正解率を得られた。今後は LSTM の前段に類似の特徴量抽出をする層を適切に追加すれば、これまでの機械学習で行っていた前処理 (特徴量の抽出) を完全に深層学習に任せながら、先行研究[3][6]に近い検知性能が達成できるのではないかと考えている。

6. まとめ

本稿ではストレージアクセスパターンを訓練データとして LSTM によるシンプルな深層学習モデルを作成し、最適なハイパーパラメータを調査した。結果として今回の訓練データでは、LSTM は 2 層、隠れ状態は 64 で最も高い正解

率を得ることを確認した。今回の深層学習モデルの正解率は、特徴エンジニアリングを施した古典的な機械学習の正解率よりも 21 クラス分類では 10% 程度劣っていた。しかし、これは時系列の情報を保持するための特徴エンジニアリングを省略したにも関わらず得られた結果であり、深層学習の 2 クラス分類の正解率は、先行研究とほぼ同等の性能 (98%) を達成した。今後、深層学習を使用したランサムウェア検知システムの性能向上を目指して前段に特徴量をうまく抽出するような層を追加することを検討する。また、訓練データにガウスノイズを追加してデータ拡張することで過学習を抑えていくことなども検討する。

謝辞 本研究は JSPS 科研費 17K00198 ならびに JSPS 科研費 20K11825 の助成を受けたものです。

参考文献

- [1] Pierazzi, F., Pendlebury, F., Cortellazzi, J., Cavallaro, L., Intriguing properties of adversarial ml attacks in the problem space, IEEE Symposium on Security and Privacy (SP), pp. 1332–1349, 2020. doi:10.1109/SP40000.2020.00073.
- [2] Berrueta, E., Morato, D., Magana, E. and Izal, M. A survey on detection techniques for cryptographic ransomware. IEEE Access, 7, pp.144925-144944, 2019.
- [3] 程田凌羽, 平野学, 小林良太郎, ストレージアクセスパターンを用いた機械学習によるランサムウェア判別システムの精度向上に関する考察, 2021-CSEC-92, 52 号, pp.1-7, 2021 年 3 月.
- [4] Hirano, M. et al. WaybackVisor: Hypervisor-based scalable live forensic architecture for timeline analysis. In International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, pp. 219-230, Springer, Cham, 2017.
- [5] Shinagawa, T., Eiraku, H. et al. BitVisor: A Thin Hypervisor for Enforcing I/O Device Security ACM VEE 2009, pp. 121-130, 2009.
- [6] Hirano, M. and Kobayashi, R. Machine Learning Based Ransomware Detection Using Storage Access Patterns Obtained from Live-forensic Hypervisor. In 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), IEEE, pp. 1-6, 2019.
- [7] 大森貴通, 水野広基, 牧原京佑, 平野学, 小林良太郎, 準バススルー型ハイパーバイザによるメモリデータ収集機能の性能改善と評価, 研究報告コンピュータセキュリティ (CSEC), 2021-CSEC-92, 2021.
- [8] Yan, B., & Han, G. Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system. IEEE Access, 6, 41238-41248, 2018.
- [9] Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., Inman, D.J., 1D Convolutional Neural Networks and Applications: A Survey. Mechanical Systems and Signal Processing 151, 107398, 2021. doi:10.1016/j.ymssp.2020.107398.
- [10] Khan, M. A., Karim, M., & Kim, Y. A scalable and hybrid intrusion detection system based on the convolutional-LSTM network. Symmetry, 11(4), 583, 2019.
- [11] Hochreiter, S. and Schmidhuber, J. Long short-term memory. Neural computation, 9(8), 1735-1780, 1997.
- [12] Garfinkel, S.L., Digital Corpora, GovDocs1, <https://digitalcorpora.org/corpora/files>, 2021 年 8 月 22 日閲覧.