Vol.2020-DPS-182 No.19
Vol.2020-CSEC-88 No.19
2020/3/12

ストレージアクセスパターンに着目した機械学習及び深層学習によるランサムウェアの検知手法の検討

程田凌羽[1]　今泉大慈郎[1]　平野学[1]　小林 良太郎[2]

概要：近年のインターネットの急速な普及により，ランサムウェアなどのマルウェアによる被害が増加している．ランサムウェアとはユーザのコンピュータにある重要なファイルを暗号化し，身代金を要求するものである．現在のランサムウェアの検知システムではシグネチャを用いたパターンマッチング方式が多く採用されている．しかし，マルウェアの作成者は既存のマルウェアに対して難読化や暗号化を施すことで，検知されにくくしている．これらの亜種のマルウェアは急速に増加しており，パターンマッチング方式ではその増加速度に追いつけない．上記の問題を解決する手法として振る舞い検知がある．振る舞い検知とは，ソフトウェアの挙動からそのソフトウェアが良性であるか悪性であるかを判断する手法である．例えば，ランサムウェアと良性のソフトウェアを比較して，ストレージ装置へのアクセスの振る舞いに違いがあれば，その特徴量を用いることでランサムウェアを検出できると考えられる．本研究ではこのストレージアクセスパターンの特徴量を用いてランサムウェアの検出を試みる．本研究ではランサムウェア３種類と，ランサムウェアと似た振る舞いをもつ良性プログラム３種類を機械学習ならびに深層学習によって分類した．本研究ではストレージの読み込みと書き込みのアクセスパターンを特徴量とする以下の２通りの学習手法を検証した．（１）単位時間あたりの読み書き量，読み書きの Logical Block Address の分散，書き込みのエントロピーからなる時系列データを Gramian Angular Field と呼ばれる手法で画像化し，畳み込みニューラルネットワークで学習させてクラス分類を行った．（２）手法（１）で用いた特徴量に加えて，読み込み量と書き込み量から求めた相関係数を新しい特徴量として追加し，従来型の機械学習である Random Forest を用いてクラス分類を行った．

キーワード：ランサムウェア，機械学習，深層学習，ストレージ装置，アクセスパターン，ハイパーバイザ

1. はじめに

企業活動や病院等の公共機関でインターネットを活用したサービスが増加しているが，それらの情報システムにおいて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ユーザの端末上の重要なファイルを暗号化して身代金を要求する悪性ソフトウェアである．身代金を支払ったとしても暗号化されたファイルがもとに戻る保証はないため，まずはランサムウェアに感染しないことが重要である．さらに，感染した場合もできるだけ早期に検知して被害を最小化し，被害から回復させることが重要である．

現状ではセキュリティ対策ソフトウェアに付属している検知機能がランサムウェアへの主な対策である．これらのランサムウェア検知機能にはシグネチャ方式が採用されている．シグネチャ方式とはランサムウェアのハッシュ値と，監視対象の実行ソフトウェアのハッシュ値が一致するかを確認する方式である．しかしシグネチャ型はランサムウェアが公開されてからハッシュ値を登録するため，ゼロデイ攻撃には対応できず，初期の被害が出る前に対策を打つことが難しい．さらに，近年はランサムウェアの亜種が急増している．これらの亜種はマルウェア自体に難読化や暗号化が施されており検知されにくくなっている．これらの亜種のハッシュ値と従来のオリジナルのランサムウェアのハッシュ値は異なっているため，新たにシグネチャを登録する必要がある．よってユーザ端末で新しい亜種を検知できるようになるまでに時間がかかる．

ランサムウェアの増加速度への対応に有効と考えられる手法が振る舞い検知である．振る舞い検知とは，検知対象のソフトウェアの挙動をみてランサムウェアであるか，正常なソフトウェアであるかを判断するものである．我々の研究室では先行研究[1][2]でストレージ装置へのアクセスパターンを利用したランサムウェアの判定と可視化のシステムを実装，評価した．これらはストレージ装置へのアクセスパターンを特徴量として機械学習をさせ，ランサムウェアが実行されたかどうかを，ストレージ装置へのアクセスパターンから検知するものである．先行研究[1][2]が特徴量として利用するストレージ装置のアクセスパターン収集には先行研究[3][4]を利用した．これは準パススルーハイパーバイザと呼ばれる軽量のハイパーバイザ[6]を利用することで，ゲスト OS を介在せずにストレージ装置へのアクセスパターンを収集するシステムである．

本研究では，上記の先行研究で実装したランサムウェア判別システムを用いて，ランサムウェア３種類，ランサムウェアに似た振る舞いをもつ良性プログラム３種類，計６種類のクラス分類問題の性能を評価した．上記の基本評価に加えて，本稿では新たに以下の２つの手法を新しく提案する．（１）先行研究の機械学習に新たな特徴量として相関係数を加える，（２）機械学習にかえてコンピュータビジョンの研究で活用されている畳み込みニューラルネットワークを時系列データの解析に利用する．とくに後者については，近年発展が目覚ましい画像認識の深層学習を利用

1 国立高等専門学校機構 豊田工業高等専門学校
National Institute of Technology, Toyota College
2 工学院大学
Kogakuin University

Vol.2020-DPS-182 No.19
Vol.2020-CSEC-88 No.19
2020/3/12

するため，時系列データを画像化する手法を採用した．以上を先行研究[1]のシステムを拡張することで実装し，ランサムウェアの検知性能を評価した．

2. ランサムウェアの挙動解析

ランサムウェアの検知手法を提案するため，まずは既存のランサムウェア検体を隔離環境で実行させ，ストレージ装置へのアクセスパターンを調査した．同様の方法でランサムウェアと似た挙動を示す良性プログラムについてもストレージ装置へのアクセスパターンを調査した．表 1 に検証したプログラムの一覧を示す．

WannaCry は2017年に世界的に猛威を振るったランサムウェアであり，CVE-2017-0144 で特定されるウィンドウズファイル共有サービスの脆弱性によって拡散する．WannaCry によってファイルは暗号化され，復号にビットコインを要求する．本研究では拡散機能は利用せず，WannaCry の検体を直接実行することによってファイル暗号化のアクセスパターンを解析した．TeslaCrypt は 2015 年頃から公開されているランサムウェアである．すでに開発者によってマスター鍵が公開されているため現在は危険のないランサムウェアであるが，代表的なランサムウェアのひとつとしてアクセスパターンを調査した．Cerber は 2016 年頃から被害が拡大した電子メールの添付ファイルで感染するランサムウェアである．ファイルを暗号化して被害者にビットコインを要求する．

ランサムウェアと似た振る舞いを持つ良性プログラムとして，Zip 圧縮プログラム，ファイルやストレージを安全に消去するユーティリティ SDelete，暗号化プログラム AESCrypt の 3 種類のアクセスパターンを調査した．SDelete は既存のファイルを各種の復元ツールを用いても復元できなくするため，元のファイル領域にデータを上書きするものである．Zip 圧縮と AES 暗号化の目的は異なるが書き込まれたデータのエントロピーが大きくなる点で似ている．

ランサムウェア検体ならびに良性プログラムはどちらも Windows 7 の 64 bit バージョンで実行した．ストレージ装置には 7,200 RPM の HDD 装置（250GB）を用いた．アクセスパターンの収集には先行研究[3]のシステムを用いた．良性プログラム３種類は以下の方法でアクセスパターンを取得した．まず GovDocs [5]と呼ばれる米国政府サイトからダウンロードしたファイルからなるデータセットから，ディレクトリ番号 000/ から 004/ までの５つのディレクトリを HDD へコピーし，それら５つのディレクトリに対して圧縮，消去，暗号化を実行した．各ディレクトリ には Word ファイル，PowerPoint ファイル，PDF ファイルなどが合計約 1,000 個含まれている．ランサムウェア３種類の実行時には，良性プログラム実行時のように上記のデータセットを HDD へコピーすることはせず Windows の標準インストールに含まれるファイルだけの HDD でランサムウェアのアクセスパターンを収集した．

表 1　検証したプログラムの一覧

	プログラム	ランサムウェア検体の SHA256 ハッシュ値（良性プログラムはバージョン等）
悪性	WannaCry	ed01ebfbc9eb5bbea545af4d01bf5f10 71661840480439c6e5babe8e080e41aa
	TeslaCrypt	3b246faa7e4b2a8550aa619f4da893db 83721aacf62b46e5863644a5249aa87e
	Cerber	e67834d1e8b38ec5864cfa101b140aea ba8f1900a6e269e6a94c90fcbfe56678
良性	Zip	Windows 7 標準プログラム
	SDelete	Version 2.02, Windows Sysinternals
	AESCrypt	Version 310, 64 bit

以上の環境で収集したストレージ装置のアクセスパターンを時系列のグラフにしたものを図 1 から図 6 に示す．なおアクセスパターンは場合によって異なるため，本稿では代表的と思われるグラフを選択した．グラフは左から，エントロピー，読み込みと書き込み量，Logical Block Address (LBA) の分散を表示している．３つのグラフはすべて横軸はプログラムを実行してからの経過時間を示しており，すべてのグラフは実行直後(0 秒)から 30 秒までの時系列データになっている．左のグラフのエントロピーは，書き込まれたセクター単位でのシャノンエントロピーを計算し，１秒間に書き込まれたすべての 4,096 byte ブロックのエントロピーを平均してプロットした．エントロピーはバイト単位で計算したため最大で 8 bit になっている[1]．エントロピーは暗号化や圧縮したデータで大きくなる傾向がある．中央のグラフは読み込み量と書き込み量を，１秒間の累積のバイト数で計算し，折れ線グラフとして表示したものである．右のグラフはストレージ装置へ読み込んだブロックならびに書き込んだブロックの番地（LBA）の平均値を１秒間隔で計算し，すべての読み込んだブロックならびに書き込んだブロックの番地（LBA）から分散を求めてプロットした．つまり，右のグラフはアクセスされた番地（LBA）のばらつきを示している．以下では，それぞれのプログラムのアクセスパターンの分析結果を示す．

2.1 WannaCry ランサムウェア

WannaCry ランサムウェアのディスク装置へのアクセスパターンを図 1 に示す．WannaCry は AES アルゴリズムによってファイルを暗号化するためエントロピーは高くなる傾向が見られた．しかしながら，WannaCry は暗号化ファイルとは別のランサムノートなどのファイルも書き込むために，エントロピーが高止まりしているとはいえない．読み込み量と書き込み量のグラフからは，読み込みのあとに書き込みが発生する規則的なパターンが確認できた．

Vol.2020-DPS-182 No.19
Vol.2020-CSEC-88 No.19
2020/3/12

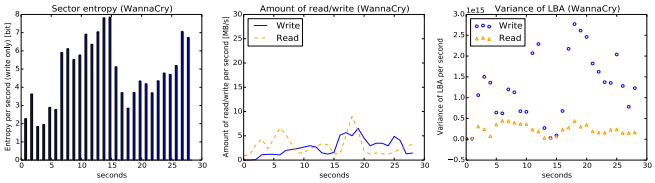

図 1　WannaCry のストレージ装置へのアクセスパターン

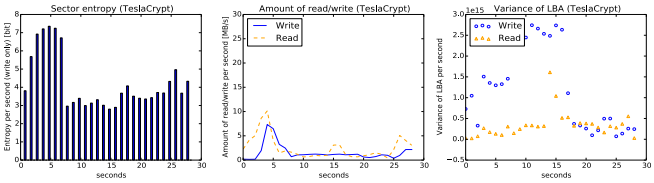

図 2　TeslaCrypt のストレージ装置へのアクセスパターン

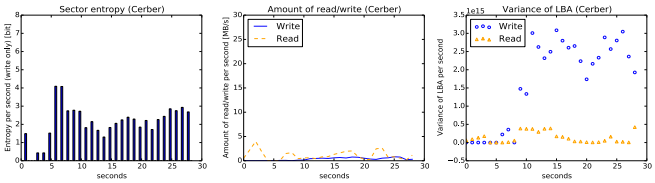

図 3　Cerber のストレージ装置へのアクセスパターン

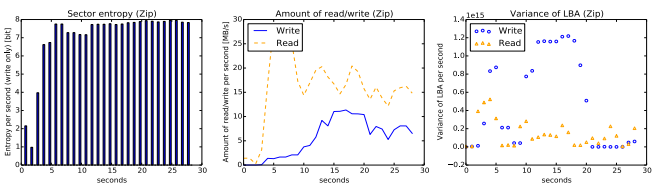

図 4　Zip のストレージ装置へのアクセスパターン

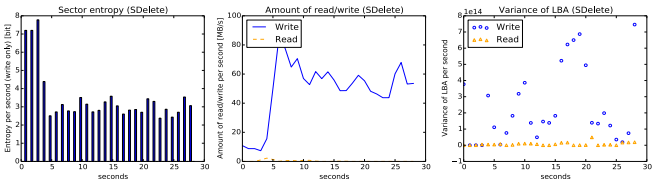

図 5　SDelete のストレージ装置へのアクセスパターン

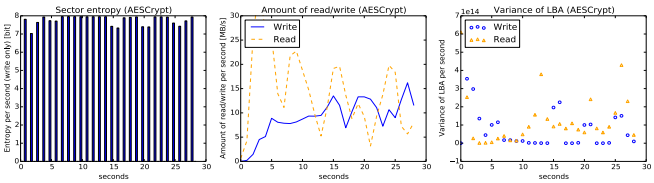

図 6　AESCrypt のストレージ装置へのアクセスパターン

LBA の分散は書き込みのほうが大きいことがわかる．WannaCry は自身のアルゴリズムに従ってファイルシステムを探索して広い範囲にファイルを書き込んでいることが推測される．

2.2 TeslaCrypt ランサムウェア

TeslaCrypt ランサムウェアのディスク装置へのアクセスパターンを図 2 に示す．TeslaCrypt は AES アルゴリズムによってファイルを暗号化する．基本的に WannaCry と似たエントロピーの変化を示しているが，読み込み量と書き込み量のパターンには特徴的な動きが見られる．

2.3 Cerber ランサムウェア

Cerber ランサムウェアのディスク装置へのアクセスパターンを図 3 に示す．エントロピーはそれほど高くなく，読み込み量と書き込み量が他の 2 つのランサムウェアと比べて少ないことがわかる．

2.4 Zip 圧縮プログラム

Zip 圧縮プログラムのディスク装置へのアクセスパターンを図 4 に示す．圧縮プログラムが書き込んだデータのエントロピーが最大値の近くで高止まりしていることがわかる．圧縮の特徴として，読み込み量に対して書き込み量が相対的に減少していることも確認できる．

2.5 SDelete 消去プログラム

SDelete 消去プログラムのディスク装置へのアクセスパターンを図 5 に示す．エントロピーは最大値の半分に満たなかった．今回は SDelete をオプションなしで実行した．SDelete は乱数でファイルを上書きした後に，ゼロで埋めるなどの処理をしているためだと考えられる．なお 図 5 の中央のグラフだけ Y 軸の最大値が 100 MB になっており，他のグラフでは Y 軸の最大値が 30 MB になっている．消去プログラムの特徴として読み込みがほとんどなく，書き込み量が相対的に大きいことが確認できる．

2.6 AESCrypt 暗号化プログラム

AESCrypt 暗号化プログラムのディスク装置へのアクセスパターンを図 6 に示す．暗号化の特徴である高いエントロピーを確認できる．エントロピーが高止まりしているのは圧縮プログラムと同様であるが，圧縮に比べて暗号化は書き込み量が若干大きいことがわかる．

3. アクセスパターンの特徴量と画像化

提案システムでは以下の二通りの方法でアクセスパターンを学習させてランサムウェアを検知する．まず学習方法（1）ではストレージ装置のアクセスパターンから特徴量を計算して機械学習をさせる．学習方法（2）ではストレージ装置のアクセスパターンの時系列データをいったん画像化して，畳み込みニューラルネットワークで深層学習させ，画像の分類問題に置き換えることでランサムウェアを検知する．

3.1 機械学習で利用する特徴量

まず学習方法（1）のために，実行直後から 30 秒間のアクセスパターンについて 10 秒間のウィンドウを時刻 0 から 20 秒までスライドさせていき，以下の 6 つの特徴量をそれぞれのウィンドウについて作成した．ウィンドウを用いて時系列の情報を含ませることを意図している．

- 書き込まれた 4,096 byte ブロックのエントロピー平均
- 読み込みされたブロックの累計バイト数

Vol.2020-DPS-182 No.19
Vol.2020-CSEC-88 No.19
2020/3/12

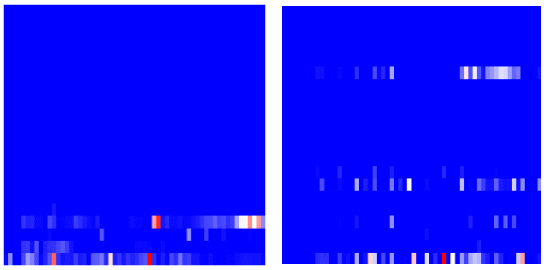

図 7　WannaCry のヒートマップ（左：Read，右：Write）

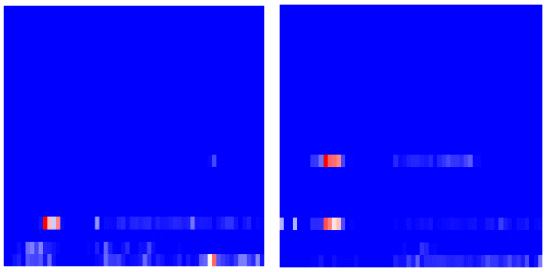

図 8　TeslaCrypt のヒートマップ（左：Read，右：Write）

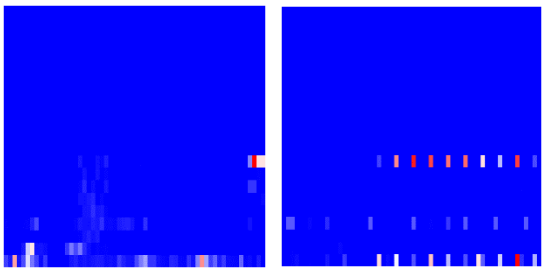

図 9　Cerber のヒートマップ（左：Read，右：Write）

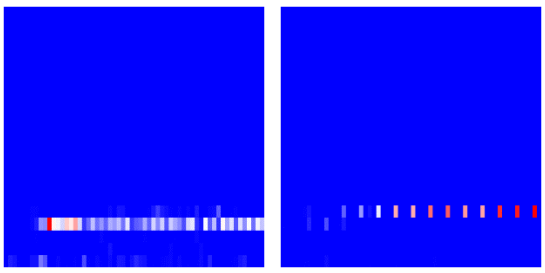

図 10　Zip のヒートマップ（左：Read，右：Write）

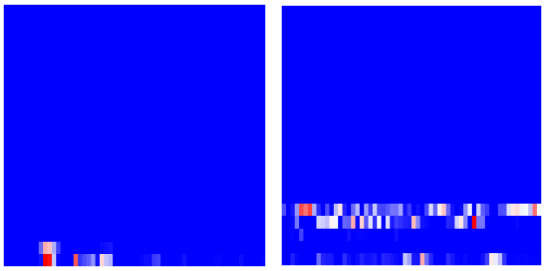

図 11　SDelete のヒートマップ（左：Read，右：Write）

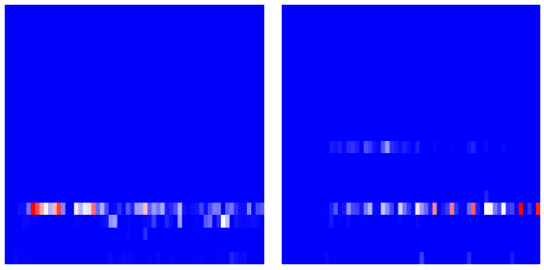

図 12　AESCrypt のヒートマップ（左：Read，右：Write）

- 書き込みされたブロックの累計バイト数
- 読み込まれたブロックの LBA の分散の平均
- 書き込まれたブロックの LBA の分散の平均
- 読み込み量と書き込み量の相関係数

最後の相関係数は本研究で新たに導入した特徴量であり，式（1）で求めた．

$$\text{相関係数 } r = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\bar{y})^2}} \qquad \text{式(1)}$$

ここで，i はウィンドウ開始からの経過時間を示しており，今回の場合は n はウィンドウサイズの 10 秒である．x は書き込み量，y は読み込み量である．

3.2 深層学習のための時系列データの画像化

学習方法（2）で画像向けの畳み込みニューラルネットワークを利用する為，本研究では二通りの画像化手法を評価した．まず，画像化手法（1）について説明する．画像化手法（1）では 30 秒間のアクセスパターンを 0.5 秒間隔で見たときに，どの LBA の範囲にアクセスがあったかをカウントし，それをヒートマップ画像にした．LBA の範囲は HDD のサイズを 20 分割して決定した．ヒートマップ画像は X 軸を時刻，Y 軸を LBA，Z 軸をアクセス回数とし，Z 軸はアクセス回数を色の濃さで表現した．この画像化手法（1）で生成したヒートマップ画像を図 7 から図 12 に示す．ヒートマップは読み込みと書き込みを別々に作成した．

次に画像化手法（2）では，Wang と Oates が提案している時系列データの画像化フレームワーク Gramian Angular Field (GAF) を用いた[7]．GAF は時系列に並べられた値を -1 から 1 の範囲にスケーリングし，それらを極座標に変換する．その後に極座標に変換された時系列データを新しい配列に変換し，それを画像として取得する．本研究では GAF のうち特に Gramian Difference Angular Field (GADF) と呼ばれる手法を採用した．変換式の詳細は参考文献[8]を参照されたい．この画像化手法（2）でも画像化手法（1）で用いた特徴量のうち相関係数を除いた値を用いた．ただし，LBA については分散を計算せずにそのまま用いた．GADF に変換前の元の配列は以下のように作成した．まず，0 秒から 30 秒まで 0.5 間隔で 60 行，LBA 範囲を（HDD 全体を 20 分割した）20 列の配列（要素数 1,200）を 3 つ作成した．60 行 20 列の個々の配列は，書き込み回数，エントロピー平均，読み込み回数について作成した．最後にそれら 3 つの配列を横に連結して 60 行 60 列（要素数 3,600）の配列を作成し，その配列を GADF 画像の配列に変換した．作成した画像を図 13 から図 18 に示す．

Vol.2020-DPS-182 No.19
Vol.2020-CSEC-88 No.19
2020/3/12

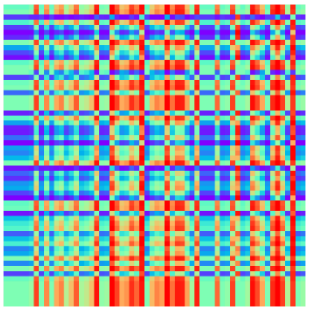

図 13　WannaCry の GADF 画像

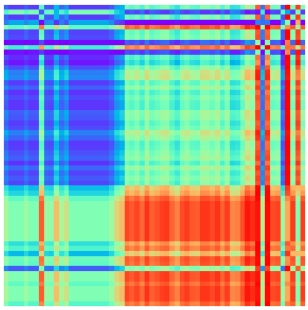

図 14　TeslaCrypt の GADF 画像

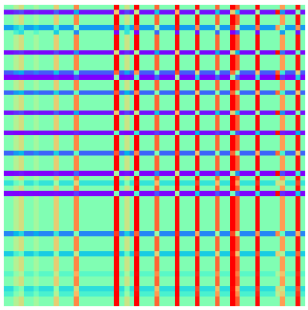

図 15　Cerber の GADF 画像

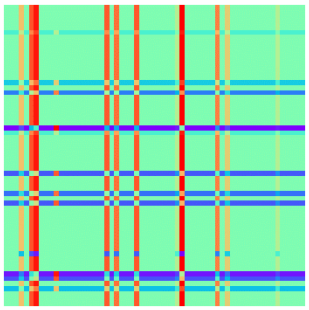

図 16　Zip の GADF 画像

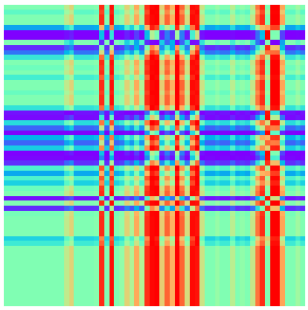

図 17　SDelete の GADF 画像

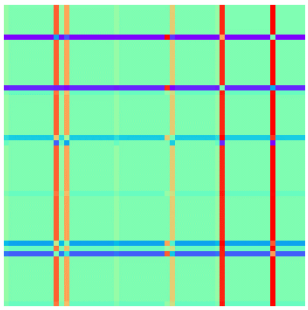

図 18　AESCrypt の GADF 画像

4. 判定システムの評価実験

学習方法（1）と学習方法（2）の二通りの方法で評価実験をおこなった．評価実験は表 1 に示した悪性のランサムウェア 3 種類と良性プログラム 3 種類の合計 6 種類を用いた．学習に利用した訓練データ数を表 2 に示す．

4.1 学習方法（1）：機械学習

学習方法（1）では，以下の 6 個の特徴量の組み合わせを変えて Random Forest で機械学習をおこない，6 クラスの分類問題の性能評価をおこなった．

A) 書き込まれた 4,096 byte ブロックのエントロピー平均
B) 読み込みされたブロックの累計バイト数
C) 書き込みされたブロックの累計バイト数
D) 読み込まれたブロックの LBA の分散の平均
E) 書き込まれたブロックの LBA の分散の平均
F) 読み込み量と書き込み量の相関係数

条件 1 では (F) の相関係数を除いた (A) から (E) までの特徴量を用いた．条件 2 では相関係数を含む (A) から (F) までの全ての特徴量を用いた．条件 3 では (B) の読み込み量と (C) の書き込み量を除いた特徴量を用いた．条件 3 の目的は (B) と (C) から求めた (F) の相関係数だけで (B) と (C) の特徴を保持できたかを確認することである．図 19 に Random Forest で求めた特徴量の寄与度を示す．

Scikit-learn の Random Forest（ハイパーパラメータは初期値のまま）を利用し，5 分割交差検証によって F 値を求めた．F 値は条件 1 のとき 0.96，条件 2 のとき 0.93，条件 3 のとき 0.89 となった．条件 2 では条件 1 より 0.03 の低下，条件 3 では条件 1 より 0.07 の低下がみられた．

表 2 訓練データの個数

	プログラム名	訓練データ数（30 秒/回）
悪性	WannaCry	13
	TeslaCrypt	11
	Cerber	12
良性	Zip	13
	SDelete	11
	AESCrypt	11

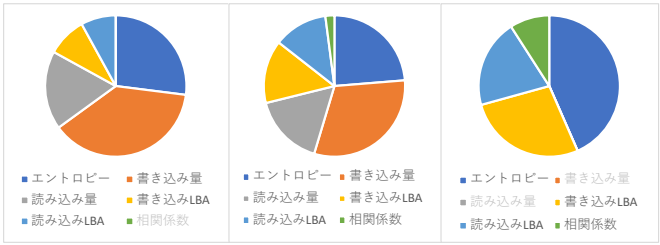

図 19　Random Forest での特徴量ごとの寄与度
（左から条件 1，条件 2，条件 3）

もうひとつの実験として，訓練データを WannaCry，TeslaCrypt，SDelete，Zip だけに限定し，作成されたモデルに対して，そのモデルにとって未知であるランサムウェア Cerber と良性プログラム AESCrypt を入力した．このとき，ランサムウェアの Cerber が学習済みのランサムウェアと認識された割合は 28%，AESCrypt が学習済みの良性プログラムと認識された割合は 70%であった．

Vol.2020-DPS-182 No.19
Vol.2020-CSEC-88 No.19
2020/3/12

4.2 学習方法（2）：深層学習

深層学習をするために 3.2 節で示した二つの画像化手法で表 2 のアクセスパターンを画像化した. 画像化手法（1）では 30 秒間で 1 回分の実行結果につき，読み込みと書き込みについて 1 枚ずつ，計 2 枚の画像を作成した．画像化手法（2）では 30 秒間で 1 回分の実行結果それぞれについて, 読み込みと書き込みをあわせた 1 枚の画像を作成した. 例えば WannaCry であれば 13 回分の実行結果があるため，画像化手法（1）では 26 枚，画像化手法（2）では 13 枚の画像を訓練データとして準備し，他の訓練データについても同様に画像を準備した．画像化手法（1）のヒートマップは Python の Seaborn ライブラリで作成した．画像化手法（2）の GADF は Python の時系列データ変換ライブラリ pyts [9]で作成した.

今回は深層学習には訓練データ数が少なすぎるため，VGG16 [10] による転移学習と ResNet50 [11] による転移学習を採用した． VGG16 と ResNet50 は ImageNet データベースで学習済みの 16 層と 50 層の畳み込みニューラルネットワークであり，どちらも 1,000 個のクラス（動物など）に分類する．ImageNet は自然物の画像が多いため，今回の幾何学模様（ヒートマップと GADF 画像）とは似ていないため高い正解率は期待できないが，今回は訓練データ数の少なさからまずは転移学習で検証することにした．VGG16 では最後の全結合層を取り除き、残りの全層の重みを固定し, 最後に新たに Keras の GlobalAveragePooling2D 平均プーリング層と Dense 全結合層を追加して学習させた．ResNet50 でも最後の全結合層を取り除き, 最後に Keras の Flatten 層と Dense 全結合層を追加して学習させた．訓練データは全体の 80%とし，残りの 20%をテストデータとして利用した．VGG16 と ResNet50 のどちらも 6 クラス分類器とした．深層学習で得られた正解率（Accuracy）を表 3 に示す．正解率が最高の 88.9%になった ResNet50 の転移学習と GADF による画像化での正解率の変化を図 20 に，損失の変化を図 21 に示す．正解率が 77.3%と 2 番目に高かった VGG16 の転移学習と GADF による画像化での正解率の変化を図 22 に，損失の変化を図 23 に示す.

表 3 深層学習の正解率

	画像化手法（1） ヒートマップ	画像化手法（2） GADF
VGG16 転移学習	56.2%	77.3%
ResNet50 転移学習	71.8%	88.9%

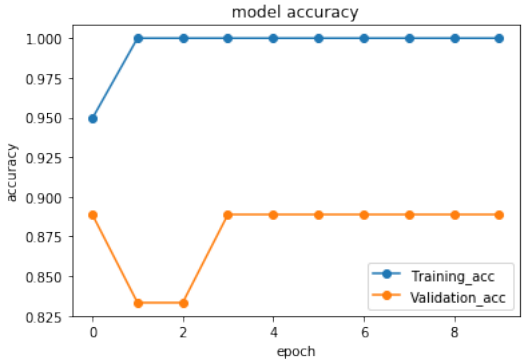

図 20 正解率の変化（ResNet50 と GADF 画像）

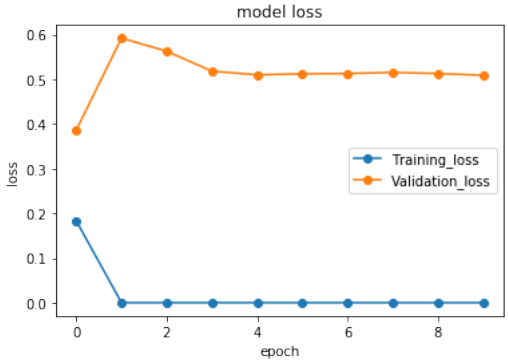

図 21 損失の変化（ResNet50 と GADF 画像）

5. 考察

まず，4.1 節の Random Forest による機械学習つまり学習方法(1)の結果について考察する．Random Forest での F 値は条件 1 が 96%と最も良好であった．F 値をさらに向上させるために読み込み量と書き込み量の相関係数の利用が効果的と考えたが，条件 2 と条件 3 の結果より相関係数は思ったほど寄与しないこと明らかになった．相関係数の寄与度は，条件 2 で 2%，条件 3 でも 9%と予想よりも低かった．他の特徴量に関しては，ランサムウェアのほうが Zip や AESCrypt よりもエントロピーが低かった．これはランサムウェアが暗号化したファイル以外のファイル（ランサムノートなど）を書き込んでいたためと考えられる．また，訓練データに含まれない未知のプログラムの検出実験では AESCrypt は 70%の割合で良性プログラム（SDelete または Zip）と認識された．これは 2 節で示したように AESCrypt のアクセスパターンが Zip と似ていたためと考えられる．しかしながら Cerber が他のランサムウェア（WannaCry と TeslaCrypt）として認識された割合は 28%と低かった．これは 2 節で示したように，Cerber のアクセスパターンが良性プログラムの SDelete と，読み込み量の少なさとエントロピーの低さに関して，似ていたためと考えられる.

次に，4.2 節の深層学習を用いた学習方法（2）の結果について考察する．まず，ResNet50 と GADF 画像の組み合わせの学習結果を見ると， 図 20 の検証データの正解率は 3 回目の epoch から変化していない．また，図 21 に関しても検証データの損失が 3 回目以降の epoch ではほとんど

Vol.2020-DPS-182 No.19
Vol.2020-CSEC-88 No.19
2020/3/12

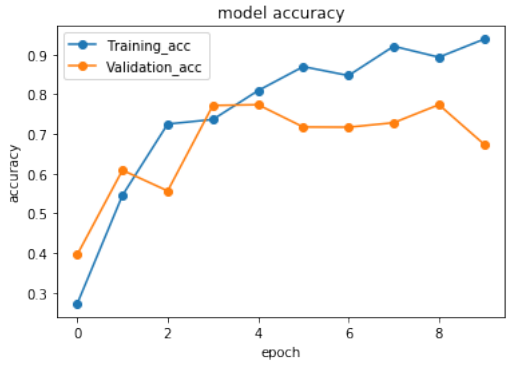

図 22　正解率の変化（VGG16 と GADF 画像）

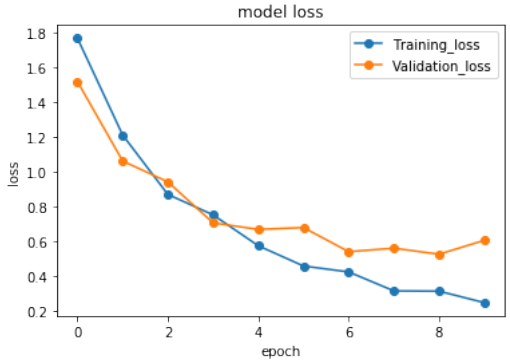

図 23　損失の変化（VGG16 と GADF 画像）

減少していない．これらの結果はモデルの評価ができるまで訓練データの数が足りていないことを示唆している．

次に，VGG16 と GADF 画像の組み合わせによる学習結果を見ると，図 22 では正解率がさきほどよりも順調に上昇しており，図 23 の損失もきれいに減少していっていることがわかった．正解率だけを見ると ResNet50 のほうが高かったものの，学習曲線をみると VGG16 のほうがうまく学習できていることがわかった．

現状では深層学習を実行するまではできた．しかし，実験結果より信頼性の高いモデルを作成するためには，たとえ転移学習であっても現状の訓練データの数を数倍から数十倍にする必要があることがわかった．このためには，アクセスパターンを収集する仕組みを自動化して訓練データを大量に収集できるようにする必要がある．また今回は用いなかった data augmentation [12] を活用した訓練データの水増しも検討する．根本的には過学習を防ぐために訓練データを十分に揃えることに加えて，訓練データに多様性をもたせる必要がある．たとえば訓練データを収集する際の監視対象となる HDD の容量を変えたり，ユーザが作成したファイルをより自然にディレクトリに配置したり，OS の種類を増やすなどして，多様なアクセスパターンを学習させる必要がある．今回は試行しなかった同じファミリーの亜種ランサムウェアについても，これから検証していくものとする．つまり，訓練データをより実環境に近い状態で多く集めることが今後の課題のひとつである．

6. おわりに

本研究ではストレージ装置のアクセスパターンを用いてランサムウェアを判別する手法を提案した．まず，アクセスパターンから特徴量を作成して機械学習させた学習方法（1）については F 値が最大で 96% の結果を得た．ただし，読み込み量と書き込み量の相関係数の導入が逆に F 値を下げる結果になってしまったので，相関係数の求め方については改善が必要なことがわかった．次にアクセスパターンの時系列データを画像化し，コンピュータビジョンのクラス分類で利用される畳み込みニューラルネットワークでランサムウェアと良性プログラムを判別する学習方法（2）を提案した．学習方法（2）については Gramian Angular Difference Field によるアクセスパターンの時系列データの画像化と ResNet50 による転移学習の組み合わせが，正解率（Accuracy）が 88.9%となり最も良好な結果を得た．VGG16 と GADF による画像化では正解率は 77.3%であったが，学習曲線をみる限り VGG16 のほうがうまく学習できていることが確認できた．今後は訓練データの数を増やして正解率ならびにモデルの信頼性を向上させたい．

謝辞　本研究は JSPS 科研費 17K00198 の助成を受けたものです．

参考文献

[1] Hirano, M. and Kobayashi, R. Machine Learning Based Ransomware Detection Using Storage Access Patterns Obtained from Live-forensic Hypervisor. 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), pp. 1-6, IEEE, 2019.

[2] 池田征士朗，髙直我，平野学，小林良太郎．ストレージアクセス履歴の時系列解析システムの実装とランサムウェア解析への応用．研究報告コンピュータセキュリティ (CSEC), 2018(10), 1-7.

[3] Hirano, M., Tsuzuki, T., Ikeda, S., Taka, N., Fujiwara, K., and Kobayashi, R. 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, Chamm, 2017.

[4] 高直我，池田征士朗，平野学，小林良太郎．準パススルー型ハイパーバイザによるストレージアクセスパターンの収集システムの提案．コンピュータセキュリティシンポジウム 2018 論文集, 2018(2), pp.678-685.

[5] Garfinkel, S., Farrell, P., Roussev, V., and Dinolt, G. Bringing science to digital forensics with standardized forensic corpora. digital investigation, 2019, 6, S2-S11.

[6] Shinagawa, T., et al. Bitvisor: a thin hypervisor for enforcing i/o device security. In Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, pp. 121-130, 2009.

[7] Wang, Z., and Oates, T. Imaging time-series to improve classification and imputation. In Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.

[8] Wang, Z. and Oates, T. Imaging time series to improve classification and imputation. arXiv 2015. arXiv preprint arXiv:1506.00327, 1506.

[9] Python package for time series transformation and classification, https://johannfaouzi.github.io/pyts/, (参照 2020-02-16)
[10] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556, 2014. (参照 2020-02-16)
[11] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.
[12] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pp. 1097-1105, 2012.