Vol.2021-DPS-186 No.52
Vol.2021-CSEC-92 No.52
2021/3/16

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

程田凌羽[1]　平野学[1,2]　小林良太郎[3]

概要：ランサムウェアの被害が増加している．ランサムウェアをはじめとするマルウェアを検知する手法には大きく分けてシグネチャ型と振る舞い型があり，商用アンチウィルス製品ではシグネチャ型が利用されていることが多い．シグネチャ型の検知システムは検体に含まれる特定のバイト列やハッシュ値を用いて検知するため，誤検知が少なく実用化に適した手法として定着している．しかしシグネチャ型の検知システムは，新種のマルウェアや亜種への対応が難しいという課題があった．そこで本研究では「振る舞い型」の検知手法に着目し，ストレージ装置へのアクセスパターンを振る舞いのモデル化に用いる．本研究で対象とするランサムウェアは被害者のファイルを暗号化して身代金を要求する．この目的を達成するためにランサムウェアはストレージ装置に保存されたファイルやディレクトリにアクセスしていき，短時間にできるだけ多くのファイルを暗号化するアルゴリズムが実装されている．本研究ではランサムウェアがストレージ装置へのアクセスパターンを隠すことは難しいという仮定のもと，それらを特徴量として採用することにした．実験では 6 個のランサムウェア検体と 3 個の無害な良性プログラムを実際に動作させ，それぞれストレージ装置へのアクセスパターンを収集した．本稿では提案方式を多様な環境に適用できるかを検証するために，HDD と SSD の 2 種類のストレージ装置，120GB と 250GB の 2 種類の容量，デスクトップに置いたおとりファイルの数とファイルサイズの条件を変えてアクセスパターンを収集した．このようにして得られたストレージ装置へのアクセスパターンを用いて機械学習モデルを訓練し，検知性能を評価した結果を報告する．最後にランサムウェアを検知する関連研究を示し，本稿で提案した検知手法について考察する．

キーワード：ランサムウェア，振る舞い，ストレージ装置，アクセスパターン，機械学習

1. はじめに

企業，官公庁，教育機関，クラウドサービス，病院等の公共サービスでのランサムウェアの被害件数が増加している．ランサムウェアとは，被害者のストレージ装置に保存してあるファイルを暗号化し，身代金を要求するような悪意のあるソフトウェアである．本稿では最初にランサムウェアをはじめとするマルウェアを検知する一般的な手法をカテゴリごとに紹介し，それぞれの利点と欠点を示し，最後に本研究の位置付けと目的を説明する．

まず，セキュリティの脅威を検知する手法には大きく分けて 2 つのアプローチがある．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．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魅力的な選択肢ではあるが，マルウェア検知に関しては多様なユーザ環境の定常状態をモデル化することが困難である．このため実際のアンチウィルス製品は 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」では，検査対象のプログラム，ファイル，通信などを，定期的に配信されてくるシグネチャと比較する．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，代表的なシグネチャにはファイルのハッシュ値や実行コードに含まれる特定のバイト列がある．しかし「シグネチャ型」の検知を迂回するために一部のコードだけを変えたり，難読化や暗号化を施した亜種も作られておりシグネチャの作成が追いつかなくなってきている．シグネチャの作成は専門の技術者が手作業で検体を解析して行うため，検体の数に比例してシグネチャを作成するコストも増大する．最近は機械学習や深層学習を用いてシグネチャを自動で作成したり，作成を支援する研究が行われている．このように「シグネチャ型」にはいくつか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．

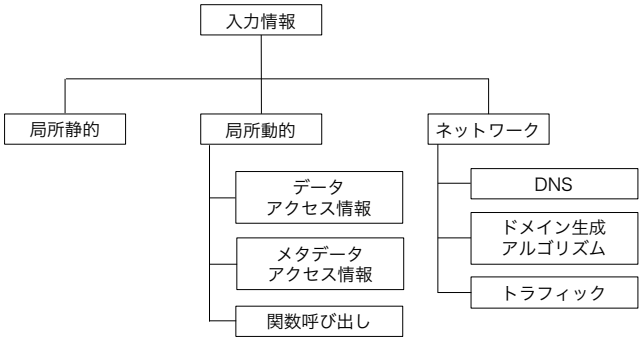

図 1 Berrueta らによるランサムウェア検知手法の分類

対して，最近は新種や亜種への対応のために「振る舞い型」の研究も盛んに行われている．「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外的な振る舞いを用いてマルウェアか正常なソフトウェアかを判定する．図 1 に

1 豊田工業高等専門学校　専攻科　情報科学専攻
2 豊田工業高等専門学校　情報工学科
3 工学院大学　情報学部

Vol.2021-DPS-186 No.52
Vol.2021-CSEC-92 No.52
2021/3/16

Berrueta らが示したランサムウェア検知システムのカテゴリ分け[1]を紹介する．Berrueta らはランサムウェアの検知システムを入力によって局所静的，局所動的，ネットワークの３つのカテゴリに分類した．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システム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などを入力としていた代わりに，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（たとえばセクタ単位でのエントロピー，読み書きされたセクタ番号の分散）を利用する．本稿では実際にランサムウェア 6 種類，無害な良性プログラム 3 種類を実験環境で動作させてストレージ装置へアクセスパターンを収集し，それらを機械学習させた結果を報告する．

2. 背景

本稿ではランサムウェアを検知するシステムの試作を報告するが，その前にストレージ装置へのアクセスパターン収集に関する先行研究を紹介し，本稿で扱うストレージ装置のアクセスパターンとは何かを具体的に説明する．

2.1 ストレージ装置からのアクセスパターン収集

まず，我々の先行研究[2]ではゲスト OS を仮想化する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．もともと Xen，VMware，KVM，Hyper-V のような汎用的なハイパーバイザは１台の物理マシンで多数のゲスト OS を動作させ，Infrastructure as a Service (IaaS) のようなクラウドサービスで複数の顧客に資源をリースしたり，データセンターにあるサーバの利用効率を高めるように設計がなされてきた．これに対し，本研究で利用する軽量ハイパーバイザ BitVisor [3] 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，アクセス制御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 だけを動作させる．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を元に近い状態で収集するのに適した設計といえる．

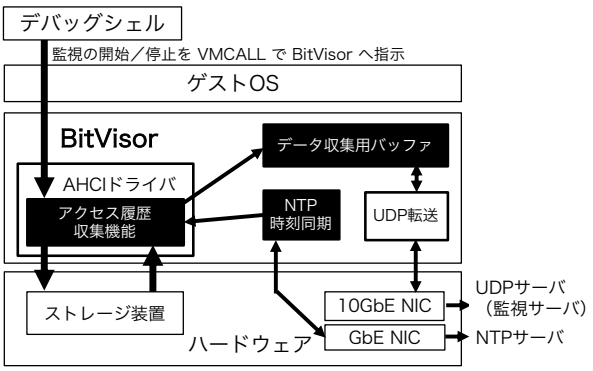

図 2 ストレージ装置へのアクセスパターンの収集

図 2 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する. 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．本稿の実験で使ったストレージ装置は１セクタが 512 バイトのものを利用した．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．

2.2 扱う問題の定義と特徴量抽出

本節ではまず我々が取り扱う問題を定義し，その後に機械学習モデルを訓練するための特徴量を定義する．本研究で取り扱う問題は基本的にクラス分類であり，ランサムウェアか良性プログラムかを判定する２値クラス分類，ランサムウェア 6 種と良性プログラム 3 種を分類する 9 クラス分類，最後にランサムウェア 6 種それぞれについて実行条件 3 つを変えた 18 種に良性プログラム 3 種を加えた 21 クラス分類である．これらの条件の詳細は後述する．

特徴量は以下のように定義した．アクセスパターンを 10 秒単位で抽出し，それらのデータから以下の値を求める．

Vol.2021-DPS-186 No.52
Vol.2021-CSEC-92 No.52
2021/3/16

- 書き込まれたデータ量（秒単位）
- 書き込まれた LBA の分散
- 書き込まれたデータのエントロピー
- 読み込まれたデータ量（秒単位）
- 読み込まれた LBA の分散

以上の 5 次元ベクトルを特徴量として定義し，それぞれに正解ラベルを付与して訓練データを作成した．これを 0 秒から 9 秒まで，1 秒から 10 秒まで，2 秒から 11 秒までと繰り返していき，それぞれの期間で 5 次元ベクトルを作成することを繰り返した．これらの特徴量は先行研究[4]で示したものであり，本稿でも同じ特徴量を用いる．本稿では先行研究と比べて，ランサムウェアの検体を追加し，実験条件を追加することで，さらに詳細に機械学習モデルを検証した．以下にその詳細を説明する．

3. 実験方法

本節では既知クラスの分類，未知クラスの分類，そして異なる OS で収集したアクセスパターンの分類の 3 つの実験方法を示す．その前にまずはランサムウェアと良性プログラムをどのような環境で実行させてアクセスパターンを収集したかを説明する．

3.1 訓練データの収集と特徴量への変換

本稿ではランサムウェアの検体として TeslaCrypt，Cerber，WannaCry，GandCrab (v4)，Ryuk，Sodinokibi の 6 種類を用い，良性プログラムとして Windows 7 にインストールされている Zip，オープンソースの暗号化プログラムである AESCrypt，Microsoft 社の Windows 向けユーティリティ集 Sysinternals スイートに含まれるデータ消去プログラム SDelete の 3 種類を用いてアクセスパターンを収集した．訓練データは，それぞれのプログラムが実行開始してから 30 秒間，60 秒間，90 秒間の 3 種類を用意した．ランサムウェアを早く検知できれば被害も少なくて済むため，検知に必要な時間を見積もるためである．

さらに，デスクトップにあるファイルの数やサイズを変えた 3 つの条件でランサムウェアのアクセスパターンを収集した．条件（1）はデスクトップにおとりファイルを置かない場合，条件（2）はデスクトップに 9,872 個のおとりファイルを置いた場合，（3）はデスクトップに 10MB 以上かつ 100MB 未満の比較的サイズの大きいおとりファイルを 605 個置いた場合である．条件（2）と条件（3）については実験が再現できるように誰でも利用可能な govdoc1 データセット[5][6]を利用した．govdoc1 データセットは米国の政府系サイトから収集したファイルのコレクションである．条件（2）は govdoc1 データセットのディレクトリ 000 から 009 に含まれるファイルを利用した．条件（3）は govdoc1 データセット全体から 10 MB 以上かつ 100MB 未満のファイルだけを抽出して利用した．

続いて，アクセスパターンの収集に用いたハードウェアを表 1 に示す．ストレージ装置として，Solid State Drive (SSD) は 2 種類，Hard Disk Drive (HDD) は 1 種類を用いた．本稿ではパーティションサイズを 120GB と 250GB の 2 種類でアクセスパターンを収集した．ただし，HDD に関しては 250GB の装置を用いて，パーティションだけ 120GB に設定して実験を行った．本稿ではそれぞれのクラスについて 2 種類のストレージ装置，2 種類のパーテーションサイズでアクセスパターンを収集した．オペレーティングシステムには 64 bit の Windows 7 Professional を利用した．

表 1　性能評価に用いたマシンの仕様

ハードウェア	規格
CPU	Intel Celeron G3920 2.9GHz (2 Core)
RAM	DDR4-2133，8GB
ストレージ装置	Seagate St250DM000 250GB HDD KingFast F6PRO 120GB SSD Crucial CT250MX 250GB SSD
マザーボード	ASRock H110M-HDV
ネットワーク	Intel X550-T1 (10GbE) Intel PRO/1000 PT (GbE)

3.2 機械学習アルゴリズムとハイパーパラメータ

本稿では，ランサムウェア 6 種，良性プログラム 3 種のそれぞれについて，2 種類のストレージ装置，2 種類のパーテーションサイズ，3 種類のデスクトップの条件，と変化させ，それぞれでプログラムを 10 回ずつ実行してアクセスパターンを収集した．それぞれの実行では最低でも 90 秒間のアクセスパターンを収集し，試行ごとにファイルに保存した．これらの収集したアクセスパターンを特徴量の 5 次元ベクトルへ変換し，機械学習モデルを訓練した．今回は機械学習アルゴリズムとして Random Forest を採用し，scikit-learn 0.23.1 の RandomForestClassifier クラスを用いて実装した．Random Forest のハイパーパラメータである木の数を 5，木の最大深さを 10 として実験した．

3.3 実験（1）既知クラスの分類

既知クラス分類実験では，ランサムウェア 6 種類それぞれについて 3 つの異なる条件（デスクトップにあるファイルの違い）で実行させて 18 クラスの訓練データを作成した．それに良性プログラム 3 種類の訓練データをあわせて，21 クラス分類をする機械学習モデルを訓練した．続いて，3 つの異なる条件の訓練データをひとまとめにし，ランサムウェア 6 種類と良性プログラム 3 種類の 9 クラス分類をする機械学習モデルを訓練した．最後に，ランサムウェアか良性プログラムかの 2 クラス分類をする機械学習モデルを訓練した．それぞれのクラス分類についてストレージ装置とパーテーションサイズ，アクセスパターンの収集時間を変化させた時の機械学習モデルの性能の変化を調査した．機械学習モデルの検証には 5-fold 交差検証を用い，指標と

Vol.2021-DPS-186 No.52
Vol.2021-CSEC-92 No.52
2021/3/16

表 2　実験（1）21 クラス分類の混同行列

			予測値																				
			Sodinokibi			TeslaCrypt			GandCrab4			WannaCry			Cerber			Ryuk			Zip	SDelete	AESCrypt
		条件	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)			
真値	Sodinokibi	(1)	75	18	8	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
		(2)	17	59	22	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
		(3)	5	10	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TeslaCrypt	(1)	0	0	0	58	29	5	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
		(2)	0	0	0	24	64	10	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
		(3)	0	0	0	9	27	51	0	0	0	0	8	0	0	0	0	0	0	1	0	0	0
	GandCrab4	(1)	0	0	0	0	0	0	20	47	35	1	0	0	0	0	0	0	0	1	0	0	0
		(2)	0	0	0	0	0	0	15	40	40	0	0	0	0	0	0	0	0	0	0	0	0
		(3)	0	0	0	0	0	1	9	29	54	0	0	0	0	0	0	0	0	1	0	0	0
	WannaCry	(1)	0	0	0	0	0	0	0	0	0	95	0	0	0	0	0	1	0	0	0	0	0
		(2)	0	0	0	0	0	0	0	0	0	0	102	0	0	0	0	0	0	0	0	0	1
		(3)	0	0	0	0	0	0	0	0	0	0	0	99	0	0	0	0	0	0	0	0	0
	Cerber	(1)	0	0	0	0	0	0	0	0	0	0	0	0	43	18	37	0	0	1	0	0	0
		(2)	0	0	0	0	0	0	0	0	0	0	0	0	21	35	51	0	1	1	0	0	0
		(3)	0	0	0	0	0	0	0	0	0	0	0	0	21	21	49	0	1	0	0	0	0
	Ryuk	(1)	0	0	0	0	0	0	0	0	1	3	0	0	0	0	0	37	34	21	0	0	0
		(2)	0	0	0	3	0	1	0	0	0	3	0	0	1	0	2	31	37	18	0	0	0
		(3)	0	0	0	0	0	0	0	1	0	0	0	0	2	1	4	27	17	63	0	0	0
	Zip		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	0	6
	SDelete		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	114	0
	AESCrypt		0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	4	0	97

して式 (1) から計算される F 値を用いた．F 値は式 (2) で求められる適合率（Precision）と再現率（Recall）の調和平均である．クラスに属する場合は Positive，属しない場合は Negative とした．

$$\text{F 値} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (1)$$

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN} \quad (2)$$

3.4 実験（2）未知クラスの分類

提案するランサムウェア検知システムでは，新種や亜種のような未知のランサムウェアへの対応能力の評価も重要である．未知クラスの分類実験では，意図的に 1 クラスを訓練データから除外し，残りの訓練データだけを用いて機械学習モデルを作成した．そして，その機械学習モデルへ訓練に使わなかったクラスを与えて推論させることで F 値を求めた．これによって，本稿の機械学習モデルが未知のプログラムをどれくらい検知できるかを評価する．実験（1）で 9 クラス分類と 21 クラス分類の検証は行っているため，実験（2）の未知クラスの分類実験ではランサムウェアか良性プログラムかだけを判定する 2 クラス分類のモデルだけを評価した．具体的には，ランサムウェア 6 種類の訓練データのうち 5 種類の訓練データだけ取り出してモデルを作成し，訓練に使わなかったランサムウェアのアクセスパターンを機械学習モデルに推論させることを繰り返した．

3.5 実験（3）異なる OS で収集したアクセスパターンのクラス分類

実験（3）ではオペレーティングシステムが異なることでランサムウェアの振る舞いが変化するか，機械学習モデルの推論にどのような影響を与えるかを調査した．3.1 節で示したように本稿では Windows 7 でランサムウェアと良性プログラムを実行してアクセスパターンを収集し，それを機械学習モデルの訓練に用いた．実験（1）と実験（2）では Windows 7 のデータだけで評価していたが，この実験（3）では Windows Server 2008 R2 を用いてテストデータを収集し，実験（1）で作った Windows 7 の機械学習モデルへ入力し，正しいクラスを推論できるかを検証した．実験では 9 クラス分類をおこなうものとし，ストレージ装置は 250GB の SSD を用い，デスクトップにおくファイルの条件に関しては 3.1 節で示した条件（2）を用い，アクセスパターンの収集時間は 90 s とした．

4. 実験結果

4.1 実験（1）既知クラスの分類

まず，21 クラス分類の混同行列を表 2 に，9 クラス分類の混同行列を表 3 に，2 クラス分類の混同行列を表 4 に示す．これら 3 つの混同行列は HDD を 120GB にパーティショニングした環境で 30 秒間収集したアクセスパターンのうち 50%を訓練データとして機械学習モデルを構築し，残り 50%のテストデータをそのモデルに入力して得た結果である．表 2 をみると WannaCry と良性プログラム 3 種に関しては条件を考慮したクラス分類ができていたが，WannaCry 以外のランサムウェアについては条件を考慮したクラス分類はうまくいかないことが多かった．しかし，ほとんどの場合に太枠で示したランサムウェアの種類ごとのクラス分類ができていることを確認できた．

次に表 3 をみると一部のランサムウェアを除いて高い精度でクラス分類できていた．この中で誤ったクラスを推論することが最も多かったのが WannaCry を入力した時であり，8% を WannaCry 以外のクラスとして推論した．WannaCry は Ryuk，GandCrab，Cerber として誤って推論さ

Vol.2021-DPS-186 No.52
Vol.2021-CSEC-92 No.52
2021/3/16

表 3　実験（1）9 クラス分類の混同行列

		予測値								
		Sodinokibi	TeslaCrypt	SDelete	GandCrab4	Cerber	WannaCry	Ryuk	Zip	AESCrypt
真値	Sodinokibi	705	4	0	0	3	0	12	0	0
	TeslaCrypt	1	664	0	2	8	6	0	0	0
	SDelete	0	0	242	0	0	0	0	0	0
	GandCrab4	0	0	0	695	13	1	23	0	0
	Cerber	5	4	0	17	684	1	5	0	0
	WannaCry	0	6	0	15	10	700	29	0	1
	Ryuk	18	1	0	5	6	8	768	0	0
	Zip	0	1	0	0	0	0	0	246	8
	AESCrypt	0	3	0	0	0	6	0	8	242

表 4　実験（1）2 クラス分類の混同行列

		予測値	
		ランサムウェア	良性プログラム
真値	ランサムウェア	1780	7
	良性プログラム	2	311

表 5　実験（1）21 クラス分類の F 値

	120GB HDD	120GB SSD	250GB HDD	250GB SSD
30秒	0.64	0.7	0.75	0.75
60秒	0.67	0.76	0.67	0.75
90秒	0.64	0.78	0.63	0.77

表 6　実験（1）9 クラス分類の F 値

	120GB HDD	120GB SSD	250GB HDD	250GB SSD
30秒	0.97	0.97	0.96	0.95
60秒	0.96	0.96	0.93	0.96
90秒	0.94	0.94	0.89	0.95

表 7　実験（1）2 クラス分類の F 値

	120GB HDD	120GB SSD	250GB HDD	250GB SSD
30秒	0.99	1	1	1
60秒	1	1	0.99	1
90秒	0.99	1	0.99	0.99

れることが多かった．他のランサムウェアも誤ってこれら 3 種類のランサムウェアに推論されることが多かったためこの 3 種には共通の特徴があると考えられる．最後に表 4 の 2 クラス分類の混同行列を見ると，最終的な目的であったランサムウェアの検知が高い精度で行われたことが確認できた．ただし本稿の訓練データはランサムウェアが多く，良性プログラムが少ないデータセットであったため，以降では各種条件で F 値を求めた結果を示して検証を進める．

まず，表 5 に 21 クラス分類の F 値を，表 6 に 9 クラス分類の F 値を，表 7 に 2 クラス分類の F 値を示す．それぞれ，ランサムウェアまたは良性プログラムを実行してからのアクセスパターンの収集時間を 30 秒，60 秒，90 秒と変えた場合，ストレージ装置の種類とサイズを変えた場合での F 値の変化を確かめた．21 クラス分類では，表 2 の混同行列で示したように，同じランサムウェアで条件が異なるだけのクラスに分類されてしまったことが原因で F 値が低下した．また，全体的に SSD のほうが HDD よりも F 値が高かった．SSD は HDD よりもアクセス速度が速いため，より多くのアクセスパターンに含まれる特徴を捉えられたと考えられる．実行時間が長くなるほど F 値が上昇すると予想していたが，9 クラス分類と 2 クラス分類では 30 秒の時が最も F 値が高くなった．これは起動直後の特徴的な動作を捉えたと考えられるが，更に検証する必要がある．

4.2 実験（2）未知クラスの分類

図 3 に未知クラスの分類実験の結果を示す．縦軸は正解率（Accuracy）を示している．この実験では 1 つのクラスを抜いてモデルを訓練し，その抜いたクラスをモデルにテストデータとして与えた結果を調査しているため，入力には正解データ（Positive）しか与えていない．よって，今回の正解率は再現率（Recall）と同じ値である．横軸は訓練で抜いたクラス（すなわちテストで使ったクラス）を示している．この実験では 2 クラス分類で正解率を求めており，正解率はランサムウェアと判定された割合となっている．実験ではストレージ装置の種類ごと（HDD または SSD），アクセスパターンを収集した時間ごとに正解率を求めた．

4.3 実験（3）異なる OS で収集したアクセスパターンのクラス分類

表 8 に Windows 7 で作成したモデルで Windows Server 2008 R2 のアクセスパターンを推論させた時の混同行列を示す．9 クラス分類問題でのマクロ平均（重みなしの単純な 9 クラスの平均値）は適合率（Precision）が 0.51，再現率（Recall）が 0.43，それらの調和平均である F 値は 0.39 であった．また，9 クラス分類ではなく，2 クラス分類として評価した場合，すなわちランサムウェアを検知する分類器として評価した場合には，適合率（Precision）は 0.89，再現率（Recall）は 0.98，F 値は 0.93 となった．

Vol.2021-DPS-186 No.52
Vol.2021-CSEC-92 No.52
2021/3/16

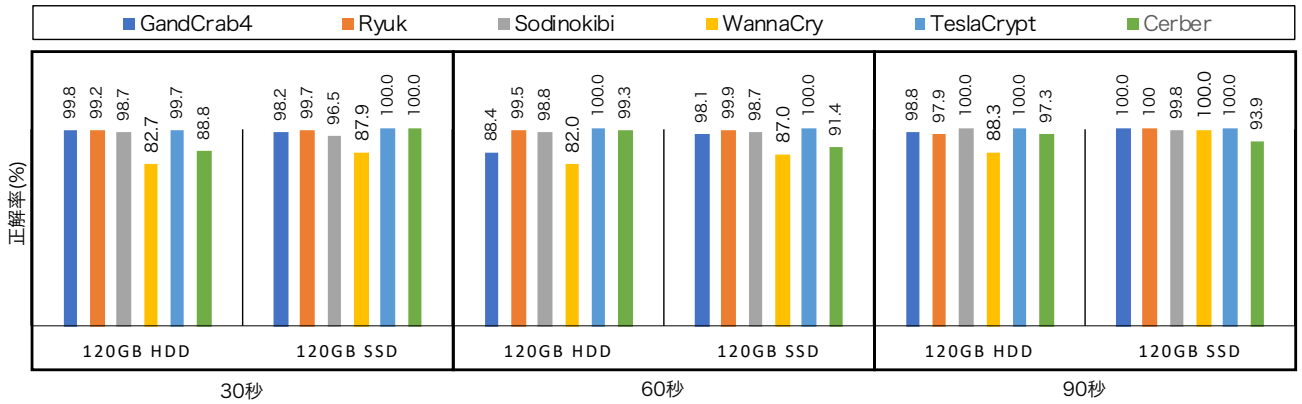

図 3 実験（2）未知クラスの分類実験の正解率

表 8 実験（3）Windows 7 のモデルで Windows Server 2008 R2 のアクセスパターンを推論させた時の混同行列

		予測値								
		SDelete	Sodinokibi	WannaCry	Cerber	Ryuk	GandCrab4	Zip	AESCrypt	TeslaCrypt
真値	SDelete	275	0	0	0	445	0	0	0	0
	Sodinokibi	0	358	146	0	184	103	4	5	0
	WannaCry	0	71	642	0	41	11	2	33	0
	Cerber	0	224	0	0	0	551	0	25	0
	Ryuk	1	126	69	0	364	191	0	8	1
	GandCrab4	0	156	0	0	164	479	0	0	0
	Zip	0	0	0	0	0	0	613	185	0
	AESCrypt	0	5	67	0	1	0	367	359	0
	TeslaCrypt	0	26	588	0	0	1	4	3	1

5. 考察

これまでに各種条件での機械学習モデルの検知性能を示してきた．本節ではランサムウェアを検知するために考慮すべき事項と課題について議論する．最後にランサムウェア検知に関する最新の関連研究をいくつか示し，本稿で示した方式と比較する．

5.1 実験（1）既知クラスの分類

既知クラスの分類実験の結果，特に 21 クラスと 9 クラスの結果より，デスクトップにあるファイルの数とサイズが変わったとしてもランサムウェアのクラス分類には大きな影響がないことを確認した．すなわち，デスクトップにあるファイルの数やサイズの差異は当初想定していたほど大きな違いを生まないことを確認できた． 次に，ランサムウェアかどうかを判定する 2 クラス分類においては F 値が概ね 1 に近い値となった． 2 クラス分類では訓練データ数がランサムウェアでは 1,787, 良性ソフトウェアでは 313 と偏りがあったものの F 値は 0.997（適合率 0.998，再現率 0.996）と良い性能を得た．また，良性プログラムをランサムウェアと誤検知した数は入力した特徴ベクトル 2,100 個のうち 7 個だけだった．つまり，目的がランサムウェアかそうでないかを判定するということであれば，本方式は誤検知の極めて少ない実用的なシステムであったといえる．

5.2 実験（2）未知クラスの分類

ランサムウェアの中から 1 つのクラスを抜き取って訓練させ，そのクラスを未知としたときの正解率（Accuracy）（この実験では Positive のテストデータしか入力していないため再現率（Recall）と同じ値）は平均 0.96 であった．ただし, WannaCry を未知のクラスとした時の正解率は平均 0.88 と，他と比べてあまり振るわなかった．この結果から言えることは WannaCry のアクセスパターンは他のランサムウェアとあまり似ていないということである．ただし，本稿で示した特徴ベクトルの作成は 10 秒ごとに区切って計算しており，30 秒のデータであれば 0 秒から 9 秒，1 秒から 10 秒,..., 20 秒から 29 秒と 21 個の特徴ベクトルに変換されて処理される．今回の場合だと，正解率 0.88 ということであるからランサムウェアの実行結果 1 回（21 個の特徴ベクトル）から考えたときに，18 個の特徴量をランサムウェアと判定したことになる．本システムを実運用する場合には 30 秒間の監視をおこなって，そのうちの閾値個以上の特徴ベクトルがランサムウェアと判定された場合には，ユーザに警告を出したり，プログラムの実行を停止させることができれば，十分に有用なシステムになると考えている．閾値を高くすると見逃しが増え，閾値を低くすると誤検知が増えるため，配備する環境が要求する安全性の度合いなどを考慮して閾値を設定する必要がある．

5.3 実験（3）異なる OS で収集したアクセスパターンのクラス分類

実験（3）で 2 クラス分類器（すなわちランサムウェアかどうかを判定する分類器）としての F 値は 0.93 であった．Windows 7 と Windows Server 2008 R2 はファイルシステムが同じ New Technology File System (NTFS) のバージョン 3.1 を使っているためストレージ装置へのアクセスパターンも似ていたと考えられる．ただし混同行列を見ると 9 クラス分類で TeslaCrypt と Cerber が正しく分類できていなかった．実際にアクセスパターンを確認してみたところ，Windows Server 2008 R2 では LBA の分散が Windows 7 より小さい傾向があった．その結果，分散以外で寄与度の高い特徴量（読み込み量とエントロピー）が似ている他のランサムウェアに分類されてしまったようである．しかし TeslaCrypt と Cerber が良性プログラムと誤って分類されることは少なかった．このように提案システムは Windows Server 2008 R2 に適用した場合でも，ある程度の検知性能を有していることを確認できた．

Microsoft 社から公式に出版されている Windows Internals [7]によると Windows 7 と Windows Server 2008 R2 はクライアント版とサーバ版の違いがあるものの，カーネルと基本的なシステムファイルは共通であることが示されている．しかしながらエディションの違いによってヒープ，スレッド，データキャッシュなどの資源割り当ての方法が異なっており，またスレッドのスケジューリングの挙動等も異なると記載されている．このようなエディションの違いによって検知性能が若干低下したと考えられる．

6. 関連研究

本節ではランサムウェアを検知する関連研究の動向を示し，本研究との比較と考察をおこなう．Berrueta らによるランサムウェア検知手法のサーベイ論文[1]によると 2015 年から 2019 年までに発表されたランサムウェアの検知に関する論文 37 件のうち 35%で機械学習が使われていたが，それ以外では経験則的な方法，たとえば特定のビットパターンの確認，カナリアファイルによる検知，Command & Control (C&C) サーバに関する挙動，DNS クエリやドメイン生成アルゴリズムに関する挙動などが使われていた．また，ファイルのエントロピーを用いて検知する研究では，カーネルドライバを用いて検知する手法[8]，検知とファイル回復の機能を備えたファイルシステム[9]，カーネルモジュールを用いて検知する手法[10] などが存在していたが，本稿のようにハイパーバイザを用いてアクセスパターンを収集した研究は調査した限り見つからなかった．

7. まとめ

本稿ではまずランサムウェアのストレージ装置へのアクセスパターンを特徴ベクトルに変換する方法を示し，ランサムウェア 6 種類と良性プログラム 3 種類で訓練データを作成し，機械学習モデルを訓練した．既知クラスの分類問題では，デスクトップにあるファイルの数や大きさが変わってもランサムウェアの検知性能は思ったほど低下しなかった．特に 2 クラス分類において F 値が 0.997 となり，高い検知性能を示した．未知クラスの分類では正解率が平均 0.964 であった．また，Windows Server 2008 R2 で実行したランサムウェアのアクセスパターンを Windows 7 で訓練したモデルに入力した結果，F 値 0.93 を得た．同時期のオペレーティングシステムであれば，ある程度は同じ機械学習モデルを流用できることを確認できた．

今後は現在の手法がランサムウェアの新種や亜種に有効かどうかを検証する．また，現在は特徴量の定義を人間が行なっているが，時系列のアクセスパターンに含まれる特徴量をすべて把握しきれているかは分からない．今後は深層学習のオートエンコーダで特徴量を自動的に抽出させたり，各種の時系列データ向け深層学習を試す予定である．

謝辞 本研究の遂行にあたり豊田工業高等専門学校情報工学科 小柳亮太氏，髙橋はな氏，遠山颯人氏にデータ収集で協力いただきました．本研究は JSPS 科研費 17K00198 ならびに 20K11825 の助成を受けたものです．

参考文献

[1] Berrueta, E., Morato, D., Magana, E. and Izal, M. A survey on detection techniques for cryptographic ransomware. IEEE Access, 7, pp.144925-144944, 2019.
[2] 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.
[3] Shinagawa, T. , Eiraku, H. et al. “BitVisor: A Thin Hypervisor for Enforcing I/O Device Security” ACM VEE 2009, pp. 121-130, 2009.
[4] 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.
[5] Garfinkel, S., Farrell, P., Roussev, V., Dinolt, G., Bringing science to digital forensics with standardized forensic corpora. digital investigation 6, S2-S11, 2009.
[6] Digital corpora, govdocs1, https://digitalcorpora.org/corpora/files, 2021 年 2 月 16 日閲覧.
[7] Mark E. Russinovich，インサイド Windows 第 6 版上，Microsoft Press（日経 BP 社），2012.
[8] Scaife N, Carter H, Traynor P, Butler KR. Cryptolock (and drop it): stopping ransomware attacks on user data. In IEEE 36th International Conference on Distributed Computing Systems (ICDCS), pp. 303-312, 2016.
[9] Continella A, Guagnelli A, Zingaro G, De Pasquale G, Barenghi A, Zanero S, Maggi F. ShieldFS: a self-healing, ransomware-aware filesystem. InProceedings of the 32nd Annual Conference on Computer Security Applications, pp. 336-347, 2016.
[10] Kharaz A, Arshad S, Mulliner C, Robertson W, Kirda E. UNVEIL: A large-scale, automated approach to detecting ransomware. In25th USENIX Security Symposium, pp. 757-772, 2016.