Computer Security Symposium 2021
26 - 29 October 2021

動的解析ログと表層情報を組み合わせたマルウェア感染活動の最終進行度推定手法

岡山 あん[1,a)]　朝倉紗斗至[1]　中川 恒[2]　押場 博光[2]　市野 将嗣[1]

概要：感染後初期の挙動から最終的にどのような被害が生じるか，またはそれに必要な対策はどのようなものかを示す “最終進行度” を推定することで，それを踏まえた対策を講じることが可能となる．本研究ではマルウェアの動的解析ログの初期の挙動と表層情報を組み合わせることで，できるだけ短いログを用いて最終進行度を推定するという手法を提案する．具体的には最終進行度を推定する分類確率が閾値未満の検体にのみ，現時点よりも感染が進行した際に得られる特徴量を回帰で予測することで，分類精度の向上と必要なログの削減を図る．実験の結果，回帰を用いないで推定する場合と比較して，最大で 2.5%の精度向上に成功した．また各検体の全てのログを学習した際に得られる精度と同程度の精度を 97.4%削減したログで実現することができた．

キーワード：マルウェア，動的解析ログ，表層情報，最終進行度推定

Final Progression Step Estimation Method of Malware Using Dynamic Analysis and Surface Logs

AN OKAYAMA[1,a)]　SATOSHI ASAKURA[1]　KO NAKAGAWA[2]　HIROMITSU OSHIBA[2]
MASATSUGU ICHINO[1]

Abstract: We are able to take measures by “Final Progression Step Estimation”, which shows the damage and needed countermeasures with the early infection behavior. In this study, we propose a method to predict the final progression step using as short logs as possible with the dynamic analysis logs and surface information. Specifically, we try to improve the classification accuracy and reduce the number of required logs by performing regression the features only on data whose classification probability is less than a threshold value. In the experiments, we succeeded in improving the accuracy by up to 2.5% when we compare the accuracy when we did not use regression. In addition, we were able to reduce 0.026 times the number of log lines with the same accuracy which we had obtained with all the logs.

Keywords: Malware，Dynamic Analysis Logs，Surface Logs, Final Progression Step Estimation

1. はじめに

マルウェア対策として侵入検知システム（IDS）がよく使われている．これはネットワーク上を流れるパケットを全て監視する機能を有するものや，定期的にログを取得，監視をし，管理者へ警告を送るものが挙げられる [1]．これを利用し，多くの場合は通知された不正なアクセス情報やパケット情報をもとに通信を遮断する．しかし近年は攻撃の巧妙化が進み，マルウェアの侵入を完全に防ぐことが難しくなってきている．そのため現在ではマルウェアの感染を前提として，感染検知後に迅速かつ正確な対応を行うことを目指す対策が主流となりつつある．マルウェアの感染を検知し攻撃を遮断した場合を考えると，この時点で得られている情報は遮断する直前までの情報であり，遮断せ

[1] 電気通信大学
The University of Electro-Communications
[2] 株式会社 FFRI セキュリティ
FFRI Security, Inc.
[a)] a.okayama@uec.ac.jp

ずに放置した際に起こりうる状況は別途分析しないとわからない．また攻撃者からの攻撃を一度遮断することに成功しても，攻撃は今後も続けて起こる可能性があるため，どの程度危険性のある攻撃が行われたのかを把握し，必要ならば対策の見直しをすることが重要である．これを実現するためには，マルウェア感染後の短い挙動のログから将来的に起こりうる被害やマルウェアの挙動を推定することが必要となる．それを踏まえ，本研究ではマルウェアの将来的に起こりうる被害やそれに対応する対策をまとめたものを “最終進行度” として定義して，これを短いログから推定することを目指す．短いログからの最終進行度推定が実現できれば，将来的にはマルウェアだけではなくクリーンウェアに対しても短いログからの推定を行うことで，マルウェア感染検知と併用した技術になると期待する．

本稿では，動的解析ログと表層情報を組み合わせた最終進行度推定を提案する．動的解析ログとは，動的解析を行う際にプログラムがホスト内部で行う挙動を記録したり，プログラムが行う通信を調査するために取得されるログデータのことである．つまり動的解析ログはマルウェアの挙動を把握するのに適しているため，最終進行度推定に利用することにする．また表層解析ログには，端末に被害が出る前に取得することができ，標的環境に影響を受けないという利点がある．そのため表層情報を活用することで，より短いログでの最終進行度推定につながると考える．

以下 2 章で関連研究，3 章で提案手法，4 章で実験方法，5 章で実験結果，6 章で考察，7 章でまとめと今後の課題を示す．

2. 関連研究

本研究では動的解析ログと表層情報を用いたマルウェアの短いログでの最終進行度推定手法を提案する．ここで，最終進行度はマルウェアの起こりうる被害やそれに対して必要な対策から定義した．そのため本章では，マルウェアのリスク分析，リスク予測に関する既存研究，マルウェアの早期推定を行っている既存研究，進行度に関連する既存研究について述べ，最後に本研究の位置づけを説明する．

2.1 リスク分析，リスク予測に関する研究

Leyla ら [4] は端末の過去一年間のバイナリファイルを解析することで，リスクレベルを定量化し，どの端末が感染の危険にさらされているのかを予測した．実験の結果，TPR が 96%，FPR が 5%の結果を得た．Kichang ら [5] は悪意のある API データベースと比較することで，Android アプリケーションのセキュリティリスクを評価する方法を提案した．実験の結果，90%以上の精度で良性，悪性アプリを分類することができた．西野ら [6] は攻撃者のネットワークへの通信試行ログを用いて，そのログが攻撃が進み深刻な被害が生じる高リスク時間帯なのか，軽微な被害にとどまる低リスク時間なのかを分類した．実験の結果，93%の精度を得た．矢野ら [7] は西野らの研究の “DNN をベースにしているために，モデルの解釈性が乏しい” という問題点を受けて，作成した DNN モデルを最もよく近似できる線形分類器を作成することで，問題の解決を図った．実験の結果，97%の精度を得た．

2.2 マルウェアの早期推定を行っている研究

Yuvraj ら [8] は静的解析を行って悪意のあるバイナリを抽出し，その後 ATT&CK [9] のフレームワークにマッピングすることで，リアルタイムで多段階の攻撃を検知する方法を提案した．実験の結果，98%の精度でマルウェアを検知することができた．Nitesh ら [10] は 4 秒の動的解析ログと，静的解析ログそれぞれにおいてマルウェアのクラス分類を行った．実験の結果，静的解析ログを用いた場合には 97%，動的解析ログを用いた場合には 99%の精度を得た．Matilda ら [11] は検体を実行してから 4 秒のログを用いて，その検体が悪意のあるものか否かを予測した．実験の結果，91%の精度を得た．

2.3 進行度に関連する研究

Sudhir ら [12] はマルウェアの進行を 12 段階に分け，各ステージにおけるマルウェアの挙動を監視することで，マルウェアか否かを分類した．実験の結果，97%の精度を得た．寺田ら [13] はマルウェアの活動を，ネットワーク通信の観点で 8 つのフェーズに整理して，それぞれのフェーズの遷移を整理した遷移モデルを定義した．その後，マルウェア通信データを各フェーズに当てはめ，最終的にマルウェアの種別を推定した．その結果，公開データを用いた場合はほぼ Gereric Trojan に誤分類したが，BOS Dataset [3] を用いた場合はすべての検体に対して正しく分類された．

2.4 本研究の位置づけ

2.1 節で述べた文献 [4], [5] は，マルウェアのリスク分析であり，短いログでの推定を想定した手法ではなかった．文献 [6], [7] は用いるデータがネットワーク通信ログである点と，研究の焦点が短いログでの推定ではない点が異なる．また 2.2 節で述べた文献 [8], [10], [11] は，必要な時間を出来るだけ短くする目的での推定である点，目的がマルウェア検知，もしくはマルウェアのクラス分類である点が異なる．2.3 節の文献 [12], [13] は，用いたデータがネットワーク通信ログである点，焦点が短いログでの推定ではない点，目的がマルウェアの検知，もしくはマルウェアのクラス分類である点が異なる．

本研究では，動的解析ログと表層情報を用いた短いログでの最終進行度推定である．

3. 提案手法

本研究の目的は，できるだけ短い動的解析ログ（以降 “短いログ” と呼ぶ）で高精度に最終進行度推定を行うことである．そのために（i），（ii）の二点工夫した．

（i）回帰により長い行数のログの特徴量を予測し利用

予備実験として，使用する動的解析ログの行数と分類精度の関係を確認したところ，使用するログの行数を長くすれば精度が高くなるということが分かった．この知見をもとに一点目の工夫として，まず長い行数のログ（以降 “長いログ” と呼ぶ）の特徴量を学習した回帰モデルで短いログの特徴量を予測し，その後それを用いて分類を行った．

（ii）表層情報の利用

表層情報は端末に被害が出る前に取得できるため，二つ目の工夫として，短いログでの推定精度を高めるために表層情報を用いた．

上記の（i），（ii）の工夫点を盛り込んだマルウェア感染後の短いログでの最終進行度推定手法を提案する．具体的には以下に示す（1）から（5）に示す手順で構成され，図示すると**図 1** である．なお本稿では，分類器が出力する各ラベルに対する所属確率を “分類確率” と定義する．

(1) 全ての検体の短いログと表層情報を組み合わせたベクトルに対して，学習，テストデータ用いて分類を行う．なおこの一回目の分類を，以降は “分類 1” と呼ぶ

(2) 分類 1 の分類確率が閾値以上の検体のみ，分類 1 の予測ラベルを正解ラベルと比較する

(3) 分類 1 の分類確率が閾値未満の検体は，長いログを学習した回帰モデルで短いログの特徴量を予測する

(4) 一定の閾値未満の検体の長いログと表層情報を組み合わせたベクトルを学習データ，回帰で短いログから特徴量を予測したログと表層情報を組み合わせたログをテストデータとして分類を行う．なおこの二回目の分類を，以降は “分類 2” と呼ぶ

(5) 分類 2 を行った検体に対して，予測ラベルと正解ラベルを比較する

分類 1 で分類確率が閾値以上の検体に関しては，分類 1 の時点で予測ラベルへの信頼度が高いと判断できるため，そのまま正解ラベルと比較する．なおこの際，あくまでも予測ラベルへの信頼度が高いと判断できるだけであるため，分類 1 のみで終わる検体の中には “予測ラベルが正しい” 検体と，“予測ラベルが正しくない” 検体の二種類が含まれている．一方で閾値未満の検体に関しては，正しい分類を行うのに情報が足りていないと判断し，特徴量を予測して情報を補ったのちに，再度分類を行い，正解ラベルと比較する．

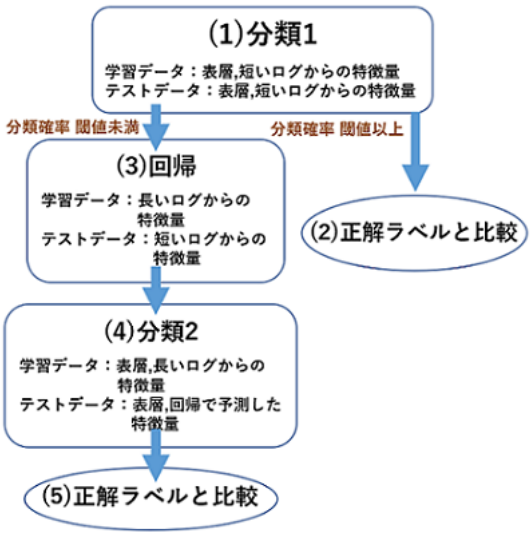

図 1 提案手法

Fig. 1 Proposed Method

4. 実験

実験方法全体を図示したものが，**図 2** である．

4.1 使用データ

実験には MWS Datasets の一部として提供される Soliton Dataset 2020 [3] を使用した．これは株式会社ソリトンシステムズのエンタープライズ向け EDR 製品である “InfoTrace Mark II for Cyber（以降 “Mark II” と呼ぶ）” が導入された環境にて取得されたログのデータセットである．Cuckoo Sandbox 上に Windows 7 Pro ベースで Mark II を導入したゲスト端末でマルウェアを実行することで，一検体につき Mark II ログ，Cuckoo ログが提供されている．また同じく Soliton Dataset 2020 には，MWS Datasets の一部として提供される FFRI Dataset と同様のツール [14] を用いて取得された表層情報も，各検体それぞれに対して提供されている．対象のマルウェアは 2019 年 1 月から 2020 年 4 月までに話題となったマルウェア 581 検体であり，調査会社などから解析結果が公開されているものを中心に VirusTotal から収集された．本研究では，全 581 検体のうち 4.2 節で説明するラベル付け方法でラベルが付与できた 554 検体を使用する．また Cuckoo ログを用いてラベル付けを行い，Mark II ログと表層情報を用いて実験を行った．

4.2 ラベル付け

Cuckoo2.0.7 には，ATT&CK の各シグネチャに対して TTPID が付与されており，これを用いてラベル付けを行った．まず ATT&CK のキーワードについて 4.2.1 項で説明し，その後 4.2.2 項で詳細なラベル付け方法を説明する．

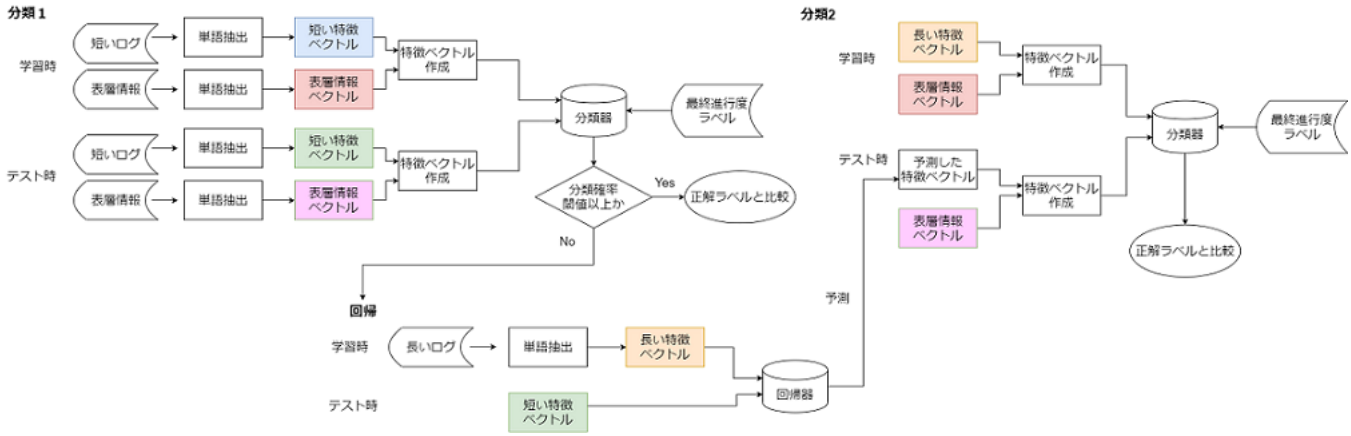

図 2　実験方法

Fig. 2　Experiment Method

4.2.1　ATT&CK

ATT&CK とは，MITRE 社が開発している攻撃者の攻撃手法および体系をまとめたフレームワーク・ナレッジベースである．これは “Adversary Group（攻撃者）”，“Software（攻撃ツール）”，“Techniques（技術）”，“Tactics（戦術）” の四つの観点でまとめられている．また ATT&CK の公式ホームページには，各 Tactics に対して用いられる具体的な Techniques がまとめてある Matrix と呼ばれる表がある．なお各 Tactics，Techniques には ID が付与されており，これを TTPID と呼ぶ．

4.2.2　ラベル付け方法

まず各検体の Cuckoo ログに付与された TTPID をすべて抽出し，それと ATT&CK の Matrix とを照らし合わせ，各 TTPID に対して適応する Tactics に変換した．次に，感染検知後の対応を行う際に考慮すべき影響範囲の観点から進行度を独自に定義した**表 1** と，各検体で出現した Tactics を照らし合わせ，付与された進行度のうち最大の進行度を最終進行度ラベルとした．つまり時間軸上での最終進行度をラベルとして扱うわけではない．そのため，初期の短いログを用いて最終進行度推定を行うにあたって，最大の進行度を示した挙動が実験に使用する初期のログに含まれる場合と，含まれない場合の二つが起こりえる．含まれる場合は，実験によって “現時点よりもマルウェアの進行が進むか否か” がわかり，初期のログに含まれない場合は，“現時点より将来的にどのくらいマルウェアの進行が進むか” がわかる．なお各最終進行度の説明は**表 2** の通りであり，ラベル付けの結果，付与されたラベルと検体数の関係は**表 3** の通りである．

4.3　実験方法

本実験で使用したアルゴリズムなどを，図 2 の順で説明していく．なお使用したパラメータは**表 4** にまとめた．また図の同じ色の特徴ベクトルは，同じ方法で作成したものを指す．

表 1　進行度ラベルの定義

Table 1　Definition of the Progression Label.

進行度	該当する Tactics
1	Initial Access, Execution, Persistence, Privilege Escalation
2	Defense Evasion
3	Credential Access, Discovery
4	Lateral Movement, Collection
5	Command and Control, Exfiltration, Impact

表 2　最終進行度の内容

Table 2　Detail of the Final Progression

最終進行度	該当する説明
1	初期感染動作で停止する可能性あり 今後攻撃が進行する可能性があるため，監視が必要
2	検知を避ける挙動をしたのち，停止する可能性あり 今後攻撃が進行する可能性があるため，監視が必要
3	アカウント情報や，環境情報が盗まれる可能性あり ID,PW を見直したり，環境を見直す必要あり
4	ファイルの構成情報の奪取や，感染の拡大を行おうとする可能性あり 重要ファイルの置き場所の再検討や，環境から感染端末を切り離す必要あり
5	C&C サーバとの通信や，重要ファイルの奪取を行おうとする可能性あり 重要ファイルの置き場所の再検討や，環境から感染端末を切り離す必要あり

表 3　ラベルごとの検体数

Table 3　Number of Data per Label

ラベル	検体数
最終進行度 1	5
最終進行度 2	264
最終進行度 3	249
最終進行度 4	31
最終進行度 5	5

4.3.1　単語抽出，特徴ベクトルの作成方法

（1）に動的解析ログ，（2）に表層情報を使用した際の単語抽出，特徴量ベクトルの作成方法を述べる．なお，動的解析ログ，表層情報を組み合わせた特徴ベクトルは，それぞれで作成した特徴ベクトルを列方向に結合したものである．

（1）動的解析ログを用いた場合

本研究では，Mark II ログからイベント (ログ中の evt)，サ

表 4　実験で用いたパラメータ
Table 4　Parameter with Experiment

	パラメータ	範囲
RF	n_estimators	100
	max_depth	6,7,8,9,10
	criterion	gini, entropy
RFR	n_estimators	16,17,18,19,20
	max_depth	13,14,15,16,17,None
DT	max_depth	7,8,9,10,11
	criterion	gini, entropy

ブイベント (ログ中の subEvt) を取り出したものをコロンでつなげたものを単語として使用した．例外として得られる挙動の情報量を増やすために，イベントが “file”，サブイベントが “close” であるとき，ファイルの読み込みのバイト数が書き込みのバイト数より多い場合は “file:read”，逆の場合は “file:write” とした．また等しい場合は，“file:close” としている．なおこれは，文献 [15] を参考に行った．次に全検体に出現する単語を網羅するコーパスを作成する．最後に各検体の Mark II ログの evt，subEvt の一組を一行と定義して，はじめから n 行目までの動的解析ログ（以降 “n 行ログ” と呼ぶ）に登場する単語からコーパスを利用して，1-gram の全単語の出現頻度を算出する．本研究では，n 行ログの単語数分の要素を持つベクトルを特徴量として用いる．なお n 行ログを使用する際に $m(< n)$ 行しかない検体に関しては，特徴量を追加せず，はじめから m 行目まで取り出した．この際も名称としては n 行ログとしている．

（2）表層情報を用いた場合

株式会社 FFRI セキュリティが提供している表層情報抽出ツールである FEXRD [17] を利用した．その結果，抽出した表層情報は 2347 種類であった．これを網羅するコーパスを作成したのち，各検体に対して 1-gram の全表層情報の出現頻度を算出し，これを特徴ベクトルとした．

4.3.2　分類 1

n 行ログから 4.3.1 項の通りにそれぞれ作成した特徴ベクトルを列方向に結合した．以降はこれを特徴ベクトルとして使用した．まず scikit-learn [16] に含まれる StandardScaler を用いて標準化し，得られた特徴ベクトルを学習データとテストデータとして用いて RandomForestClassifier（以降 “RF” と呼ぶ）で分類を行った．なお分類は，学習データとテストデータのラベルの比率をそろえるために StratifiedKFold を用いて，層化 5 分割交差検証で行い，学習データとテストデータに使用した n 行ログは，同じ行数のものを使用した．また RF のパラメータを表 4 の範囲で，GridSearch で全特徴量 2374(MarkII : 27，表層情報 : 2347) 次元の特徴量に対して最適なパラメータを探した．そして SelectFromModel で重要度が中央値以上である特徴量を最大 46 個抽出し，その後，選択した特徴量ベクトルに対して再度 GridSearch を用いて最適なパラメータを探した．なお本研究ではパラメータチューニングの指標に，StratifiedKFold を用いた層化 3 分割交差検証で RF を用いて分類を行った際の accuracy を用いた．また分類時には RF に用意されている predict_proba を用いて予測ラベルの分類確率を算出し，それが閾値以上であった検体は，正解ラベルと比較した．閾値としては，55，60，65%を使用した．

4.3.3　回帰

4.3.2 項の分類確率が閾値未満であった検体は動的解析ログの特徴量を回帰で予測した．以下，詳細な方法を説明する．まず StandardScaler を用いて標準化したのちに RandomForestRegressor(以降 “RFR” と呼ぶ) を用いて，StratifiedKFold の層化 5 分割交差検証で短いログの特徴量を予測した．回帰を行う際，学習データとして短いログ，テストデータとして長いログを用いた．パラメータチューニングには平均二乗誤差が用いられる場合が多いが，マルウェアには冗長の操作を行う検体も多いため，平均二乗誤差の値が大きくても分類に不要な挙動情報をなくした分類に適した特徴量になっている可能性も考えられる．本研究の回帰の目的はあくまで，回帰後に行う分類の精度向上である．そのため本研究では，回帰のパラメータチューニングの指標として，4.3.2 項の分類を RF の代わりに DecisionTreeClassifier（以降 “DT” と呼ぶ）を用いて StratifiedKFold の層化 4 分割交差検証で行った際の accuracy を用いた．なお動的解析ログから作成した特徴ベクトルは，4.3.1 項（1）のように独自にすでに特徴量を選択しているといえるため，この分類中では SelctFromModel で特徴量選択は行わない．また分類時の学習データは長いログ，テストデータは短いログから回帰で特徴量を予測したログを用いた．

4.3.4　分類 2

回帰を用いて特徴量を予測した後，再度分類を行った．なおこの際の学習データは長いログと表層情報を組み合わせた特徴量ベクトル，テストデータは短いログから特徴量を予測したログと表層情報を組み合わせた特徴量ベクトルである．また使用した分類方法，パラメータ等は 4.3.2 項と同じものである．

4.4　評価方法

提案手法の有効性を以下の二つの観点で評価した．

(1) 学習に用いる短いログを固定として，回帰のテストに用いるログの行数を増やしていく

(2) テストに用いる長いログを固定として，回帰の学習に用いるログの行数を減らしていく

（1）は短いログを使用したときの精度向上を目的に行った．例として短いログを 15 行，長いログを 20 行使用する場合を考える．15 行のログを使用して全検体に対して分類 1 回のみで最終進行度推定をした精度を，一部検体に対して 15 行のログの特徴量を予測して 20 行のログとした提案手法

表 5　動的解析ログのみを用いた予備実験
Table 5　Preliminary Experiment with Dynamic Analysis Logs

使用した行数	1	10	20	30	40	50	...	全行数使用
分類精度 (%)	49.8	57.0	57.9	61.4	85.6	87.4	...	91.9

表 6　動的解析ログ, 表層情報を用いた予備実験
Table 6　Preliminary Experiment with Dynamic Analysis and Surface Logs

使用した行数	1	10	20	30	40	50	...	全行数使用
分類精度 (%)	84.1	84.7	84.8	87.9	89.0	90.4	...	92.1

の精度が上回ることができれば，一部検体に回帰を適用することによって精度向上が見込めたといえる．

（2）は早期の最終進行度推定を意識して必要なログの削減を目的に行った．例として短いログを 30 行，長いログを 1000 行使用する場合を考える．このとき 1000 行のログを使用して全検体に対して分類 1 回で最終進行度推定を行った精度と，提案手法の 30 行の精度が等しければ，1000 行必要な精度を 30 行で得ることができたといえる．このときの削減率は，$(1000-30)/1000 = 0.97\%$ と計算する．

5. 実験結果

5.1 予備実験

提案手法を実装する前提として，用いる動的解析ログを長くすることで分類精度が高くなるか確認するため，動的解析ログのみを使用した分類実験と，表層情報と組み合わせた分類実験を行った．

用いた n 行ログの n を “使用した行数”，それを使用して実験を行った際に得られた正答率を “分類精度” として，動的解析ログのみを用いた分類結果を**表 5**，表層情報と組み合わせた分類結果を**表 6** にまとめた．

5.2 動的解析ログのみを用いた実験

動的解析ログと表層情報を組み合わせる有効性を示すために，4.3 節の提案手法の実験を動的解析ログのみで行った．なお動的解析ログから作成した特徴ベクトルは，4.3.1 項（1）のように独自にすでに特徴量を選択しているといえるため，分類中では SelctFromModel で特徴量選択は行わない．

4.4 節の通り，提案手法の評価には二種類の設定で実験を行う．まず精度向上をみる一つ目の設定では，閾値を 55，60，65%，短いログとして 10，15，20 行のログで実験をした際，最も精度がよかったケースは閾値を 55%，短いログとして 20 行を使用した結果であった．このときの詳しい結果を実験結果を**表 7** にまとめる．ベースとしている 20 行の精度は分類 1 回で精度評価を行った際の精度であり，これを超えると提案手法によって回帰を用いることで精度向上がみられたといえる．ここで該当する精度を太字にしている．表 7 をみると，太字の精度のうち 80 行の 58.5%の時が最も高いことがわかる．つまりこれは，提案手法によってベースとなる 20 行の精度 57.9%から，最大 0.6%精度向上がみられたことがいえる．

またログの削減割合をみる二つ目の実験設定では，閾値を 55，60，65%，長いログとして各検体すべてのログを使用したもので実験をした．この際，最もログを削減できたケースは閾値を 55%にした場合であり，この結果を**表 8** にまとめる．なお一つ目の実験設定とは異なり，分類 1 を行うログ，つまり短いログが固定ではないため正解数 1，正解数 2 の分母である “各分類を行った検体数” は異なる．ここで 91%以上の精度を出したものを各検体すべての行数を使用した場合の精度と同等であるとすると，要件を満たす最も短い行数は 80 行とわかる．本実験で使用した Soliton Dataset 2020 の 554 検体の総ログ行数の平均は，2679.5 行である．ログの全行数を使用する場合と 80 行のみを使用する場合では同等の精度が出ているといえるので，ログの削減割合は，$(2679.5-80)/2679.5 \fallingdotseq 97.0\%$ と計算できる．

5.3 提案手法の実験

同様に 4.3 節の実験を，4.4 節の二種類の設定で実験を行った．

精度向上を見る一つ目の実験設定では，閾値を 55，60，65%，短いログとして 10，15，20 行のログで実験をして，精度向上を確認する．その結果最も精度がよかったケースである閾値が 65%，短いログとして 10 行を使用した際の詳しい結果を**表 9** にまとめる．5.2 節と同様，ベースとなる 10 行の 84.7%を超えている行は精度向上が見られるため，太字で示している．表より，80 行の 87.2%が最も精度が高いといえる．よって最大で 2.5%の精度が向上したといえる．

次に二つ目の実験設定でログの削減割合を確認する．閾値を 55，60，65%，長いログとして各検体すべてのログを使用したもので実験をした．この際，最もログを削減できたケースは閾値を 55%にした場合であり，この結果を**表 10** にまとめる．5.2 節と同様に，91%以上の精度を出したものを各検体すべての行数を使用した場合の精度と同等とみなすと，要件を満たす最も短い行数は，70 行が当てはまる．ここで 5.2 節と同様にログの削減割合を計算すると，$(2679.5-70)/2679.5 \fallingdotseq 97.4\%$ とわかる．

6. 考察

本節では，最終進行度推定に表層情報を用いたことによる有効性，短いログから特徴量を予測したことによる有効性の二つの観点から実験結果を考察する．

6.1 表層情報を用いたことによる有効性

はじめに精度向上に表層情報がどの程度寄与したかを考察する．Mark II は，初期段階のログではプロセスの起動を

表 7　動的解析ログのみを用いた最終進行度推定 1
Table 7　Final Progression Estimation 1 with Dynamic Analysis Logs

使用した行数	20	30	40	50	60	70	80	90	100	110
分類 1 の正解数 /275		166	166	166	166	166	166	166	166	166
分類 2 の正解数 /279		154	107	122	139	149	158	156	150	153
分類精度 (%)	57.9	57.8	49.2	52.0	55.1	56.9	**58.5**	**58.1**	57.0	57.6
使用した行数		120	130	140	150	160	170	180	190	200
分類 1 の正解数 /275		166	166	166	166	166	166	166	166	166
分類 2 の正解数 /279		156	156	156	155	156	156	156	156	156
分類精度 (%)		**58.1**	**58.1**	**58.1**	**58.1**	57.9	**58.1**	**58.1**	**58.1**	**58.1**

表 8　動的解析ログのみを用いた最終進行度推定 2
Table 8　Final Progression Estimation 2 with Dynamic Analysis Logs

使用した行数	10	20	30	40	50	60	70	80	90	100	
分類 1 の正解数	131	166	308	463	468	473	490	501	492	490	
分類 2 の正解数	179	136	33	9	14	16	11	4	6	8	
分類精度 (%)	56.0	54.5	61.6	85.2	87.0	88.3	90.4	**91.2**	89.9	89.9	
使用した行数	200	300	400	500	600	700	800	900	1000	...	全行数使用
分類 1 の正解数	488	490	487	488	486	486	488	494	494		
分類 2 の正解数	11	14	13	13	11	14	12	10	8		
分類精度 (%)	90.1	91.0	90.3	90.4	89.7	90.3	90.3	91.0	90.6	...	91.9

多くの検体が行っており，用いる行数を増やすほどレジストリやファイルへの操作が増え，ラベルごとに差が出てくるという特徴があった．実際これは予備実験で用いる行数を増やすほど精度が向上していることからもわかる．しかし本研究では，できるだけ短いログを用いることに焦点を当てている．ここで**表 11** に表層情報から抽出できた特徴量の中で，マルウェア検知において有用だとされる file_size の値の平均を最終進行度ラベルごとにまとめた．これを見ると，file_size 情報のみでもラベルごとに差異が見えることがわかる．実際に表層情報のみを用いた分類実験を行ったところ，最終進行度ラベル 2，3 の他に，最終進行度 4 の検体を 31 検体中 26 検体分類することができた．このように表層情報はラベル間の差異が，短い動的解析ログよりもみられることがわかったため，本研究ではできるだけ短い動的解析ログから高精度で推定するために表層情報を組み合わせることにした．実際の結果である表 7，表 9 をみると，動的解析ログのみを用いた場合だと全体的に 57〜58%という精度であったのに対し，表層情報と組み合わせることで全体的に 86〜87%に精度を大きく向上させた．また表層情報のみを用いた実験では 83.8%の精度を得たため，動的解析ログと表層情報を組み合わせることで，最大 3.4%の精度向上を得たといえる．

次にログの削減に表層情報がどの程度寄与したか考察する．5.2，5.3 節を見ると，動的解析ログのみを用いた実験では 97.0%，表層情報を組み合わせた実験では 97.4%ログを削減することができたことがわかる．これより，表層情報を取り入れたことによって 0.4%削減できたといえる．

以上の点から，表層情報と動的解析ログを組み合わせることは，精度向上に大きく寄与し，ログの削減に寄与した．

6.2　短いログから特徴量を予測したことによる有効性

はじめに回帰を用いて短いログから特徴量を予測したことが，どの程度精度向上に寄与したか考察する．表 9 で最大の精度を得た 10 行のログを回帰で 70 行まで特徴量を予測した場合を例として考える．この際，2.5%の精度向上に成功したことがわかっている．結果について分析すると，分類 1 で不正解，分類 2 で正解であった検体は 33 検体であり，逆に分類 1 で正解，分類 2 で不正解であった検体は 21 検体であった．この差である 12 検体分が，分類 1 で精度評価をせず分類 2 で精度評価を行ったことによって，分類結果が改善されたマルウェアに該当する．一方で提案手法では，最終進行度 1，5 の多くを正しく分類することができなかった．ログを見ると最終進行度 1，5 と他ラベルとの差異が生じてくるのは大体 40 行ほどであり，回帰を用いて 70 行まで特徴量を予測したこの実験では本来正しく分類できるはずだが，実際は回帰による予測精度が不十分であり，分類することができなかった．

次に回帰を用いて短いログから特徴量を予測したことが，ログの削減にどの程度寄与したかを考察する．5.3 節では，70 行使用した精度が全行数を使った分類精度と同等であったため，97.4%ログを削減することに成功したと述べた．Mark II ログの平均行数は 2679.5 行であり，これを 70 行で同程度推定できたのは意義があったと考える．

以上の点から，短いログから特徴量を予測したことは，精度向上に寄与し，またログの削減に大きく寄与したといえる．

7.　まとめと今後の課題

本稿では，短いログでの最終進行度推定を，分類確率が閾値未満の検体にのみ回帰を適用する手法で提案した．実

表 9　動的解析ログ, 表層情報を用いた最終進行度推定 1

Table 9　Final Progression Estimation 1 with Dynamic Analysis and Surface Logs

使用した行数	10	20	30	40	50	60	70	80	90	100	110
分類 1 の正解数 /453		415	415	415	415	415	415	415	415	415	415
分類 2 の正解数 /101		64	62	64	64	62	64	68	65	67	65
分類精度 (%)	84.7	**86.5**	**86.1**	**86.5**	**86.5**	**86.1**	**86.5**	**87.2**	**86.6**	**87.0**	**86.6**
使用した行数		120	130	140	150	160	170	180	190	200	
分類 1 の正解数 /415		415	415	415	415	415	415	415	415	415	
分類 2 の正解数 /101		66	65	65	65	66	66	66	66	63	
分類精度 (%)		**86.8**	**86.6**	**86.6**	**86.6**	**86.8**	**86.8**	**86.8**	**86.8**	**86.3**	

表 10　動的解析ログ, 表層情報を用いた最終進行度推定 2

Table 10　Final Progression Estimation 2 with Dynamic Analysis and Surface Logs

使用した行数	10	20	30	40	50	60	70	80	90	100	
分類 1 の正解数	446	436	462	469	476	486	497	495	493	491	
分類 2 の正解数	20	29	19	23	19	14	15	11	13	13	
分類精度 (%)	84.1	83.9	86.8	88.8	89.4	90.3	**92.4**	91.3	91.3	91.0	
使用した行数	200	300	400	500	600	700	800	900	1000	...	全行数使用
分類 1 の正解数	493	497	496	499	497	496	496	500	499		
分類 2 の正解数	16	10	13	10	14	12	15	9	10		
分類精度 (%)	91.9	91.5	91.9	91.9	92.2	91.7	92.2	91.9	91.9	...	92.1

表 11　表層情報の特徴

Table 11　Features of Surface Logs

最終進行度	file_size の数値の平均
1	169260
2	7503782637880
3	2647880
4	299388
5	1939906

験の結果，回帰を行わず分類 1 回で行った精度よりも最大で 2.5%向上した．また各検体の全てのログで分類した場合と同程度の精度を得るのに必要なログを，97.4%削減することに成功した．

今後はさらにデータ数を増やした実験を行い，提案手法の有効性を確認していく．また本研究の段階ではマルウェア検知の付加情報としての活用を想定していたが，将来的にマルウェア検知を兼ねる技術に発展させることを期待している．そのために今後はクリーンウェアデータを含めた実験を行っていく．

参考文献

[1] 竹下隆史，他: マスタリング TCP/IP 入門編第 5 版，オーム社，2018，p.326

[2] Cuckoo:Cuckoo Sandbox 2.0.7,https://cuckoosandbox.org/blog/207-interim-release（2021/08/08 参照）.

[3] 寺田真敏，他: マルウェア対策のための研究用データセット MWS Datasets ～コミュニティへの貢献とその課題～，情報処理学会，Vol.2020-IFAT-139 No.8（2020）.

[4] Leyla，Bilge. et al.: RiskTeller: Predicting the Risk of Cyber Incidents. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. pp.1299-1311（2017）.

[5] Kichang, Kim. et al.: Risk Assessment Scheme for Mobile Applications Based on Tree Boosting. IEEE Access, Vol. 8, pp.48503-48514.(2020)

[6] 西野琢也，他: テンソル分解に基づくグラフ分類による組織内ネットワーク攻撃活動検知，コンピュータセキュリティシンポジウム（2017）.

[7] 矢野正太郎，他: 組織内ネットワーク攻撃進行度の自動推定技術の評価検証，第 80 回全国大会講演論文集，pp.453-454（2018）.

[8] Yuvraj, Sanjayrao Takey. et al.: Real Time early Multi Stage Attack Detection, 2021 7th International Conference on Advanced Computing and Communication Systems(2021).

[9] MITRE:ATT&CK，https://attack.mitre.org/（2021/08/08 参照）.

[10] Nitesh，Kumar. et al.: Malware Classification using Early Stage Behavioral Analysis. 2019 14th Asia Joint Conference on Information Security（2019）.

[11] Matilda，Rhode. et al.: Early-Stage malware prediction sing recurrent neural networks. Computers&Security. Vol.77, pp.578-594（2018）.

[12] Sudhir，Kumar. et al.: A Lifecycle Based Approach for Malware Analysis. 2014 Fourth International Conference on Communication System and Network Technologies（2014）.

[13] 寺田成吾，他: 通信挙動に基づくマルウェア種別分類手法，コンピュータセキュリティシンポジウム（2017）.

[14] GitHub：ffridataset-scripts,https://github.com/FFRI/ffridataset-scripts/releases/tag/v2020.1(2021/08/08 参照).

[15] 朝倉紗斗至，他: 動的解析ログを用いた特徴量の予測によるマルウェアの早期機能推定に関する検討，コンピュータセキュリティシンポジウム (2020).

[16] scikit-learn：scikit-learn,https://scikit-learn.org/stable/(2021/08/08 参照).

[17] GitHub：FFRI/FEXRD,https://github.com/FFRI/FEXRD(2021/08/08 参照).