ログ間の特徴量予測によるマルウェアの目的推定に関する検討

篠崎 佑馬[1,a)]　中川 恒[2]　茂木 裕貴[2]　押場 博光[2]　市野 将嗣[1]

概要：近年，マルウェアを用いたサイバー攻撃は巧妙化しており，これらによる侵害の検知が難しくなっている．マルウェアの侵入後には，その影響範囲を調査し，システムの復旧および将来的な攻撃に対する防御策を講じる必要がある．この影響範囲の調査において，侵害に使用されたマルウェアの目的を知ることは有用であると考えられる．本稿では，日常的な収集が可能であるイベントログを用いて，マルウェアの目的を推定することに着目する．そこで，マルウェアの実行時に得られたイベントログの特徴量から，サンドボックス解析などで得られる情報量の多い API コールログの特徴量を予測し，それを使用してマルウェアの目的を推定する手法を提案する．イベントログには InfoTrace Mark II ログ，API コールログには Cuckoo ログを用いて分類実験を行った結果，イベントログで目的推定を行う場合よりも，本手法を用いた場合の方が高精度で推定が行えることが確認された．

キーワード：マルウェア，目的推定，特徴量予測，インシデントレスポンス

A Study on Objective Estimation of Malware by Feature Prediction between Logs

Yuma Shinozaki[1,a)]　Ko Nakagawa[2]　Yuki Mogi[2]　Hiromitsu Oshiba[2]　Masatsugu Ichino[1]

Abstract: In recent years, cyber attacks using malware have become more sophisticated, making it difficult to detect intrusions caused by these attacks. After malware invades a system, it is necessary to investigate the extent of its impact and take measures to restore the system and defend against future attacks. When investigating the extent of the impact, it is considered useful to know the objective of the malware used to invade the system. In this paper, we focus on estimating the objective of malware using event logs, which can be collected on a daily basis. We therefore propose a method to predict the features of API call logs, which contain a lot of information and can be obtained by sandbox analysis, from the features of event logs obtained when malware is executed, and use them to estimate the objective of malware. As a result of a classification experiment using InfoTrace Mark II logs for event logs and Cuckoo logs for API call logs, it was confirmed that the estimation was more accurate when using this method than when using event logs.

Keywords: Malware, Objective estimation, Feature Prediction, Incident Response

1. はじめに

近年，マルウェアはますます巧妙化しており，従来の防御手法では対応できない場合もある [1]．マルウェアに感染した場合はその被害に対する，インシデントレスポンスという対応を取る必要がある．インシデントレスポンスでは，被害を受けたシステムの保護，影響範囲や被害の特定，システムの復旧，次回の攻撃に備えた対策などを行う [2]．

インシデントレスポンスにおいて，影響範囲の特定や次回の攻撃に対する防御策を講じる際には，マルウェアが行う侵害活動の目的を把握することが有効だと考える．一般

[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)] y.shinozaki@uec.ac.jp

にマルウェアはその侵害活動に目的を持っており，それには他のマルウェアのダウンロードや，データの暗号化などといったものがある [3]．例えばマルウェアの目的が情報窃取だと分かれば，機密ファイル群が標的となったと推測できるため，機密ファイル群を中心に影響範囲の調査を行い，それらの防御を強化する，などといった対応を取ることができる．またマルウェアは暗号化と情報窃取など複数の目的を持つこともあり，目的を見落とすことなく把握することで，適切な対応につなげることができる．

インシデントレスポンスにおけるインシデントの調査では，主に OS やアプリケーションの動作を記録したイベントログが使われる [2]．このようなイベントログは，ログサイズが比較的小さく，長期保存が可能であるが，記載できる情報量は多くない．一方，サンドボックス解析を行うことで取得できる API コールログは，イベントログよりも低レイヤの情報を記録できるため情報量が多いが，ログサイズも大きい．したがって，インシデントレスポンスに備えて API コールログを常に保存しておくことは難しい．

マルウェア対策としては侵入検知システム（IDS）がよく使われている．IDS によって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，インシデントレスポンスに使用できるログはマルウェアの実行が停止するまでの短い挙動のログとなる．したがって，限られた情報でマルウェアの目的を推定する必要がある．

そこで本稿では，短い期間の挙動のイベントログを用いて，より高精度でマルウェアが持つ複数の目的を推定する手法を提案する．具体的にはイベントログの特徴量から API コールログの特徴量を予測し，それを使用してマルウェアの目的を推定する．ログサイズの小さいイベントログと，情報量の多い API コールログの両方の利点を活かすことができれば，インシデントレスポンスにおいて有用な情報を得ることができると考える．

本研究の貢献は以下のとおりである．

(1) 異なるログ間（イベントログと API コールログ）の特徴量予測による目的推定手法を提案した

(2) 高精度かつインシデントレスポンスに役立つ，複数目的を持つマルウェアにも対応した目的推定を行った

本稿では，2 章にて関連研究，3 章は提案手法，4 章で実験，5 章で実験結果，6 章では結果の分析と考察，最後に 7 章にてまとめと今後の課題を議論する．

2. 関連研究

2.1 マルウェア分類に関連する研究

Pektaş ら [4] は，Windows API コールログを用いて，N-gram と Voting Experts アルゴリズムによって特徴を抽出することでマルウェアのファミリ分類を行った．Kumar ら [5] は，マルウェアの表層情報とマルウェアの 4 秒間の動的解析ログからマルウェアのクラス分類を行った．Catak ら [6] は，Windows API コールログの新しいデータセットを作成し，LSTM モデルを用いてマルウェアのクラス分類を行った．Aslan ら [7] は，マルウェア検体のバイナリファイルをグレースケール画像に変換することで，ファミリ分類を行う手法を提案した．Xu ら [8] は，Windows API コールログを用いてマルウェアの Behavior tree を構築し，ファミリ分類を行った．Chen ら [9] は，マルウェアのバイナリファイルとアセンブリファイルを用いた，マルウェアのファミリ分類手法を提案した．

2.2 マルウェアの目的推定に関連する研究

森ら [10] は，攻撃者の攻撃目的推定に向け，イベントログを用いてマルウェアの特徴的な挙動の分析およびマルウェアの役割推定を行った．Kawaguchi ら [11] は，感染初期の Windows API コールログを用いてマルウェアの機能推定を行った．複数のアルゴリズムで分類を行い，Random Forest などの決定木ベースのアルゴリズムでは比較的高精度で推定できることを示した．児玉ら [12] は，Windows API コールログを用いてマルウェア間の類似度を利用した機能推定手法の検討を行った．N-gram 及び LCS によって導出した類似度による機能推定がともに有効であることを示した．Mandiant の FLARE チームは，静的解析を通してマルウェアのファイル特徴量と逆アセンブリ特徴量を抽出し，事前に定義されたルールを元にマルウェアの機能を検知することができるオープンソースソフトウェア CAPA[13] を作成した．Islam ら [14] は，マルウェアの API コール列と ATT&CK techniques のマッピングを手動で作成し，Malware Deception に役立つフレームワークの提案を行った．Sun ら [15] は，マルウェアの実行ファイルを逆コンパイルすることでアセンブリ命令と API コール列の特徴量を抽出し，その ATT&CK techniques を推定した．Graph Convolutional Networks (GCN) と ATT&CK knowledge graph を用いて，マルウェアと ATT&CK techniques の対応を学習した．Feng ら [16] は，IoT マルウェアの Function Call Graph から，その悪性動作を ATT&CK Tactics に基づいて分類した．マスクされたグラフ表現を用いてノイズを除去することで，高精度で推定ができることを示した．

2.3 本研究の位置づけ

本研究の目的は，インシデントレスポンスに役立つマルウェアの目的推定である．本節では，インシデントレスポンスでの適用にあたり，目的推定に求められる要件について整理する．主な要件は以下の 4 点だと考える．

(i) 日常的な収集，長期保存が可能である動的解析ログを用いて目的推定できる

(ii) 途中で攻撃を遮断した状況を考慮し，マルウェアの実行が途中で停止するまでに得られる短い期間の挙動のログから，マルウェアの目的をより高精度で推定できる

(iii) 推定結果からマルウェアの目的を明確に読み取れる
(iv) マルウェアが複数の目的を持つ可能性を考慮した推定ができる（暗号化 + 情報窃取など）

以上を踏まえ，関連研究と本研究を比較する．関連研究 [4], [7], [8], [9] は，マルウェアのファミリ分類を行っている．ファミリ分類の結果からは，マルウェアの目的を明確に読み取ることができない．

関連研究 [5], [6] における推定結果は，ファミリ分類よりはマルウェアの目的を読み取りやすいが，推定に用いているログが長期保存に向かない API コールログである点，マルウェアが複数の悪性動作を行うことを考慮していない点が本研究と異なる．関連研究 [11], [12] は，マルウェアのマルチラベル機能推定を行っているが，これらも推定に用いているログが API コールログである点が本研究と異なる．関連研究 [10] は，イベントログを用いて役割推定を行っているが，マルウェアが複数の悪性動作を行うことを考慮しておらず，感染初期での分類を前提としていない．

関連研究 [13], [14] は，機械学習を用いない方法でマルウェアの機能を推定している点が本研究と異なる．事前に大量のルールを作成する必要があり，情報収集をすることには多くの時間と労力がかかる．

関連研究 [15], [16] は，マルウェアの実行ファイルを分析し静的な特徴を用いている点が本研究と異なる．マルウェアの静的な特徴は難読化やパッキングによって変化するため，静的な特徴を用いる研究では精度に影響があるが，本研究で用いる動的解析ログはそれらの影響を受けない．

以上より関連研究は本研究と異なる点があり，4 つの要件全てを満たしているものはない．そこで本研究では上記の要件全てを満たすため，短い期間のイベントログでのマルウェアのマルチラベル目的推定を行うことを目指す．

3. 提案手法

本稿では，短い期間の挙動のイベントログを用いた特徴量予測によるマルウェアの目的推定手法を提案する．提案手法では，イベントログの特徴量から API コールログの特徴量を予測した後，その予測した特徴量を用いて目的推定を行う．2.3 節で示した要件を満たすため，提案手法では以下の 3 点について工夫した．

(1) 短い期間のイベントログから API コールログの特徴量を予測し使用
まず，イベントログは日常的に収集可能かつ長期保存可能な動的解析ログであるため，イベントログを用いた目的推定を行うことで要件 (i) を満たす．ここで，イベントログと API コールログでは情報量に大きな差があるため，ログ間の回帰を行うことで情報量を増やすことができると考えられる．また，用いるイベントログよりも長い記録時間の API コールログを用いることで，マルウェアが感染後しばらく経過してから行う悪性動作も学習することができる．したがって，ログに記録される情報とマルウェアの実行時間という 2 つの側面から，イベントログよりも情報量が増加することが期待される．以上より，短い期間のイベントログからより高精度な推定が可能となると考えられ，要件 (ii) が満たされる．

(2) マルウェアの目的を想定した分類ラベルを使用
要件 (iii) を満たすため，マルウェアファミリを分類ラベルとせず，マルウェアが行う主な悪性動作を元に侵害箇所の推定に役立つ目的ラベルを定義した（本稿で示す実験では “download”, “encrypt”, “spread”, “infostealer”, “mining”, “RAT”, “drop” の 7 種を定義し使用）．

(3) マルチラベル分類を実施
1 つのデータが複数のラベルを持つことを許容する，マルチラベル分類のアルゴリズムで目的推定を行った．これによりマルウェアが複数の悪性動作を行う可能性を考慮した推定が可能となるため，要件 (iv) が満たされる．

以上の工夫点を踏まえた提案手法の構成を**図 1** に示す．まず，Random Forest 回帰器にイベントログの特徴ベクトル，API コールログの特徴ベクトルを入力し，イベントログの特徴量から API コールログの特徴量を予測する回帰器を学習させる．次に，Random Forest 分類器に元の API コールログの特徴ベクトルと，回帰器によって予測された特徴ベクトルの 2 種を入力し，ログの特徴量から目的ラベルを推定する分類器を学習させる．このとき，予測した特徴ベクトルも分類器の学習に用いることで，学習データ数を増やすことができるため，分類器の精度向上に繋がると考えられる．回帰器・分類器の学習後，未知のイベントログを回帰器に入力し，予測ベクトルを得た後にマルチラベル分類することで，目的ラベルを取得し目的推定を行う．

4. 実験

4.1 データセット

4.1.1 使用データ

MWS Datasets[17] の一部として提供されている，Soliton Dataset 2020, 2021 を使用した．Soliton Dataset では，1 検体につき Mark II ログ（イベントログ）と Cuckoo ログ（API コールログ）が提供されている．

本研究では，十分な情報量を含む Cuckoo ログを抽出するため，Cuckoo ログのログサイズが 1MB 以上かつ 4.1.2 節でラベル付けした検体 723 検体を使用した．

4.1.2 マルウェアの目的ラベル

関連研究 [18] を参考に，マルウェアの目的ラベルを定義した．本研究で設定したラベルは，“download”, “encrypt”, “spread”, “infostealer”, “mining”, “RAT”, “drop” の 7 種類である．これらのラベルを，セキュリティベンダが公開している脅威レポート等を参考に，検体ごとに付与した．1 検体が複数の悪性動作を行う場合は，それに当てはまるような複数のラベルを付与した．実験に用いた 723 検体の

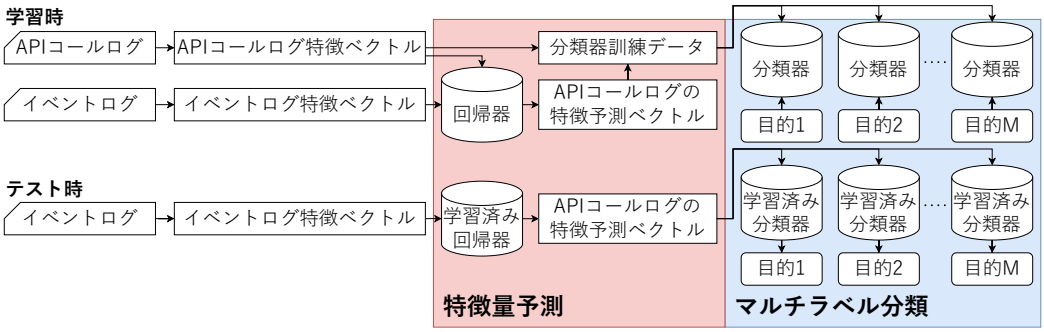

図 1　提案手法の構成
Fig. 1　Configuration of the proposed method

表 1　マルウェアの目的ラベルごとの検体数
Table 1　Number of samples for each malware objective label

download	encrypt	spread	infostealer
210	82	87	266
mining	RAT	drop	
14	176	189	

目的ラベルごとの検体数を**表 1** に示す.

4.2　特徴量の抽出

Mark II ログ（イベントログ）と Cuckoo ログ（API コールログ）から複数カテゴリの単語を抽出し，それらの単語の出現頻度もしくは出現有無を特徴量とした.

4.2.1　Mark II ログ（イベントログ）

Mark II ログは，攻撃を途中で遮断した状況を考慮し，提供されたログから最初の 100 挙動のみを切り出したログを使用した．その後，関連研究 [18] を参考に以下 2 つのカテゴリの特徴量を抽出した.

(1) イベント: Mark II ログに記録されたイベント（ログ中の “evt”）とサブイベント（ログ中の “subEvt”）をコロン（“:”）で連結したものを単語として抽出し，そのイベントの出現頻度を特徴量とした．（単語例: “ps:start”）(2) ファイル: マルウェアによって扱われたファイル名を抽出し，そのファイルの出現有無を特徴量とした．ログ中から，“<ファイル名>.<拡張子>” の形式の文字列を単語として抽出した．（単語例: “WmiPrvSE.exe”）

4.2.2　Cuckoo ログ（API コールログ）

Cuckoo ログは，あらかじめ検体を解析して手に入れたログだと想定し，全時間分のログを使用した．関連研究 [5] を参考に以下 8 つのカテゴリの特徴量を抽出した.

(1) Windows API: API コール列を抽出し，それぞれの Windows API の出現頻度を特徴量とした．（単語例: “NtCreateUserProcess”）(2) Signature: Cuckoo Sandbox はマルウェア固有の動作を “signature” として検出する．これらの signature を単語として抽出し，出現有無を特徴量とした．（単語例: “One or more processes crashed”）(3) ドメイン: Cuckoo Sandbox によって抽出された IoC の URL、その他マルウェアが解決しようとしたドメイン名を抽出し，その出現有無を特徴量とした．（単語例: “gmail.com”）(4) プロセス: Cuckoo ログに記録されたプロセス名を抽出し，その出現有無を特徴量とした．（単語例: “schtasks.exe”）(5) ファイル: マルウェアによって読み書き，作成，削除されたファイル名を抽出し，その出現有無を特徴量とした．（単語例: “desktop.ini”）(6) プロセス数，ドロップファイル数: Cuckoo ログに記録されたプロセスと，ドロップされたファイルの数を計上し，その値を特徴量とした．(7) DLL ファイル: ロードされた DLL ファイル名を抽出し，その出現有無を特徴量とした．（単語例: “netutils.dll”）(8) レジストリキー: マルウェアによって開かれた，または読み書きされたレジストリキーを抽出し，その出現有無を特徴量とした．（単語例: “AutoRun”）

4.3　特徴ベクトル作成

ログから抽出したすべてのカテゴリの特徴量を結合し，特徴量選択を行うことで，それぞれのログの特徴ベクトルを作成した．4.4 節で示す通り層化 10 分割交差検証を行っているため，特徴量選択は各交差検証ごとに行った.

特徴量選択では，まず各カテゴリにおいて，訓練データ中に 2 回以上出現した単語を選択した．次に選択された単語をすべて結合した後，Random Forest の重要度に基づき，Recursive Feature Elimination（RFE）により特徴量を選択した．このとき，イベントログ特徴ベクトルは 300 次元，API コールログ特徴ベクトルは 500 次元となるように特徴量選択を行った．最後にイベントログ，API コールログ特徴ベクトルを Z-Score Normalization により標準化し，これを最終的な特徴ベクトルとした.

4.4　機械学習モデルの学習と評価

以下では，回帰器と分類器の学習と評価のために Nested cross validation を実施した．Outer loop では，層化 10 分割交差検証によって訓練&検証データとテストデータに分割し，テストデータを使って実験の評価を行った．Inner

表 2　API コールログでの分類
Table 2　Classification of API call logs

	download	encrypt	spread	infostealer	mining	RAT	drop	average
Accuracy	0.909	0.952	0.990	0.908	0.974	0.880	0.982	0.942
TPR	0.914	0.790	0.956	0.944	0.450	0.926	0.958	0.848
FPR	0.093	0.028	0.005	0.113	0.017	0.135	0.010	0.057

表 3　イベントログでの分類
Table 3　Classification of event logs

	download	encrypt	spread	infostealer	mining	RAT	drop	average
Accuracy	0.859	0.845	0.926	0.748	0.957	0.753	0.887	0.854
TPR	0.829	0.340	0.769	0.763	0.000	0.807	0.846	0.622
FPR	0.129	0.091	0.052	0.261	0.024	0.263	0.099	0.131

loop では，訓練&検証データをさらに 3 分割交差検証によって訓練データと検証データに分割することで，機械学習モデルのチューニングを行った．

4.4.1　特徴ベクトルの予測

回帰器を用いてイベントログ特徴ベクトルから API コールログ特徴ベクトルへの予測を行った．回帰器の機械学習アルゴリズムには Random Forest を用い，イベントログ特徴ベクトルを説明変数，API コールログ特徴ベクトルを目的変数とした．回帰器のハイパーパラメータは，予測した特徴ベクトルと API コールログ特徴ベクトルの平均二乗誤差（MSE）が最小となるように，Optuna[19] によるベイズ最適化にてチューニングした．

4.4.2　目的分類

API コールログ特徴ベクトルを説明変数，マルウェアの目的ラベルを目的変数とするマルチラベル分類器を用いて目的分類を行った．分類器の機械学習アルゴリズムは Random Forest を用い，元の API コールログ特徴ベクトルと予測した特徴ベクトルの両方を用いて学習を行った．

マルチラベル分類のアルゴリズムは Binary Relevance（BR）[20] を用いた．これは各目的ラベルごとに二値分類器を学習し，その結果を結合する手法である．したがって本実験では，4.1.2 節の 7 種類の目的ラベルの有無を判定する二値分類器を 7 つ学習した．分類器のハイパーパラメータは，TPR が最大となるように，グリッドサーチを用いて各目的ラベルの分類器ごとにチューニングした．

4.4.3　評価

目的ラベルごとに，以下の式 (1)〜(3) で定義した Accuracy, True Positive Rate (TPR), False Positive Rate (FPR) とそれらのマクロ平均を算出した．

$$\mathrm{Accuracy} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{1}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \quad \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} \tag{2}$$

$$\mathrm{Average}_{\mathrm{Metric}} = \frac{1}{M}\sum_{i=1}^{M}\mathrm{Metric}_i \tag{3}$$

M は目的ラベルの数である．式 (3) において，Metric には Accuracy, TPR, FPR のいずれかが入る．TP, TN, FP, FN はそれぞれ，検体が各目的ラベルを持っている場合を陽性，持っていない場合を陰性としたときの，真陽性，真陰性，偽陽性，偽陰性の数を表す．

また複数の目的を持つ検体に対して，マルチラベル分類において部分的な正解も考慮することができる評価指標 [21] で評価を行った．N 検体中 i 番目の検体の予測ラベルを Y_i，正解ラベルを Z_i とし，部分点を考慮した $\mathrm{Accuracy}', \mathrm{Recall}', \mathrm{Precision}'$ を以下の式 (4)〜(6) で定義する．なお $|Y_i| = 0$ となる検体 i の $\mathrm{Precision}'$ は 0 とした．

$$\mathrm{Accuracy}' = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \tag{4}$$

$$\mathrm{Recall}' = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap Z_i|}{|Z_i|} \tag{5}$$

$$\mathrm{Precision}' = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap Z_i|}{|Y_i|} \tag{6}$$

5.　実験結果

5.1　目的ラベルごとの評価

API コールログでの結果を**表 2**，短い期間のイベントログでの分類結果を**表 3**，短い期間のイベントログから回帰器によって予測した API コールログを用いて分類した際の結果を**表 4** に示す．表の各行はそれぞれの目的ラベルに対する Accuracy，TPR，FPR である．

まず，表 2 と表 3 を比較すると，API コールログを用いた方が全ての指標で高い精度が得られていることがわかる．したがって，イベントログよりも API コールログを用いた方がマルウェアを正確に分類できるため，特徴量予測によりイベントログの分類精度が向上することが期待される．表 3 と表 4 から，予測した API コールログを用いた方が各指標のマクロ平均が高いことが読み取れる．特に TPR においては，encrypt 検体では 8.8%，マクロ平均は 3.4% の精度向上が確認された．したがってイベントログで目的

表 4　予測した API コールログでの分類
Table 4　Classification of predicted API call logs

	download	encrypt	spread	infostealer	mining	RAT	drop	average
Accuracy	0.861	0.842	0.926	0.767	0.960	0.752	0.907	0.859
TPR	0.824	0.428	0.781	0.752	0.150	0.801	0.857	0.656
FPR	0.125	0.105	0.054	0.225	0.024	0.263	0.075	0.124

表 5　マルチラベルを考慮した評価
Table 5　Evaluation considering multi-label

	Accuracy′	Recall′	Precision′
イベントログ	0.854	0.745	0.754
提案手法	0.856	0.773	0.760

推定を行う場合よりも，本手法を適用することでより高精度で目的の推定を行うことができるといえる．イベントログ特徴量から API コールログ特徴量に予測を行い，また短時間から長時間のログを予測したことで，短い期間のイベントログの情報量を補完することができたと考えられる．

また，実験に用いたイベントログのログサイズは 1.2GB，API コールログのログサイズは 32GB であった．本提案手法により，API コールログを用いて目的推定を行う場合と比べてログサイズを約 96%削減することができた．

5.2　マルチラベルを考慮した評価

複数の目的を持つ 214 検体について，4.4.3 節で示したマルチラベルを考慮した評価の結果を**表 5** に示す．表 5 から，Accuracy′, Recall′, Precision′ ともにイベントログよりも提案手法での分類の方が高精度であることが読み取れ，特に Recall′ は約 3%向上していることがわかる．マルウェアの目的を完全でなくとも，部分的に推定できればインシデントレスポンスの補助として役立つと考えられる．したがって，複数の目的を持つ検体に対しても本提案手法による分類が有効であると考えられる．

6.　考察

6.1　予測性能に関する評価

分類性能の向上にログ間の特徴量予測がどのように寄与したかについて考察する．特徴量予測を高精度で行うことができた場合，元の API コールログ特徴量によって正しく分類された検体は，予測した API コールログ特徴量を用いた場合でも正しく分類できるようになると考えられる．したがって本節では，(a) 元の API コールログ特徴量によって正しく分類され，予測した API コールログ特徴量でも正しく分類できた検体 (b) 元の API コールログ特徴量によって正しく分類されたが，予測した API コールログ特徴量では誤って分類された検体 について議論する．

各目的ラベルにおける検体 (a) と (b) に対して，それらの検体数および，元の API コールログ特徴量と予測した API コールログ特徴量の間の平均二乗誤差を求め，その結果を**表 6** に示した．表 6 から一部の検体の分類が誤っていることがわかり，どの目的ラベルにおいても検体 (b) よりも (a) の方が平均二乗誤差が小さいことが読み取れる．また特徴量の可視化を行うため，検体 (a) と (b) から例として download, encrypt, RAT の目的ラベルを持つ検体を 1 検体ずつ抽出し，それらの Windows API 頻度特徴量の一部を抜き出してヒストグラムとして表したものをそれぞれ**図 2** と**図 3** に示した．図 2 と図 3 の横軸は特徴量，縦軸は特徴量の値となっており，青色の棒は予測した API コールログ特徴量，赤色の棒は元の API コールログ特徴量である．つまり，青色と赤色の棒が類似していれば予測が高精度で行われたということである．図 2 においては青色と赤色の棒が類似しており，図 3 においてはそれらに大きく差が生じていることが読み取れる．以上の平均二乗誤差と可視化の結果から，特徴量予測が高精度で行えるほど分類性能が向上することが確認でき，予測性能を上げることができれば分類性能も向上する可能性があると考えられる．

次に，特徴量予測を行ったが誤って分類された検体（検体 (b)）について考察する．ここで，図 2 と図 3 に示した検体の API 呼び出しの合計数を**表 7** に示す．表 7 から，図 2 の検体よりも図 3 の検体の方が API 呼び出しの合計数が多いことが読み取れる．そして，図 3 の検体は一部の特徴量が他の特徴量と比較して大きくなっており，予測した特徴量が実際の特徴量よりも極端に小さくなっている．つまり，図 3 の検体は図 2 の検体に比べて一部の API が非常に多く呼ばれており，それらの API 量を正しく予測できていないといえる．検体 (b) の他の検体においても特徴ベクトルを調査したところ，複数検体で図 3 と同様の傾向が見られ，一部の API が他の検体と比べて非常に多く呼ばれている検体が存在した．したがって，一部の API が非常に多く呼ばれている検体の特徴量予測では，それらの API 量を実際よりも極端に少なく予測してしまう場合があり，これは予測した API コールログ特徴量を用いた分類における精度低下の一因になっていると考えられる．

また表 6 から，encrypt や RAT の目的ラベルでは特徴量予測の平均二乗誤差が特に大きいことが読み取れる．上記の調査において，encrypt と RAT の一部の検体では，一部の API が他の検体と比べて特に多く呼ばれており，予測した API 量と実際の API 量の差が，図 3 と比べて非常に大きくなっていることを確認した．したがって，特徴量予測

表 6　検体 (a) と検体 (b) の検体数と平均二乗誤差
Table 6　Number of samples and mean squared error for sample (a) and sample (b)

	検体数		平均二乗誤差	
	検体 (a)	検体 (b)	検体 (a)	検体 (b)
download	169	23	0.371	2.705
encrypt	29	36	69.148	1640.471
spread	68	15	0.124	0.724
infostealer	197	54	0.261	0.827
mining	0	7	-	2.290
RAT	131	32	3.669	22756.309
drop	162	19	0.127	1.795

表 7　API 呼び出しの合計数
Table 7　Total number of API calls

	download	encrypt	RAT
図 2	8807	11646	17786
図 3	155259	108916	192532

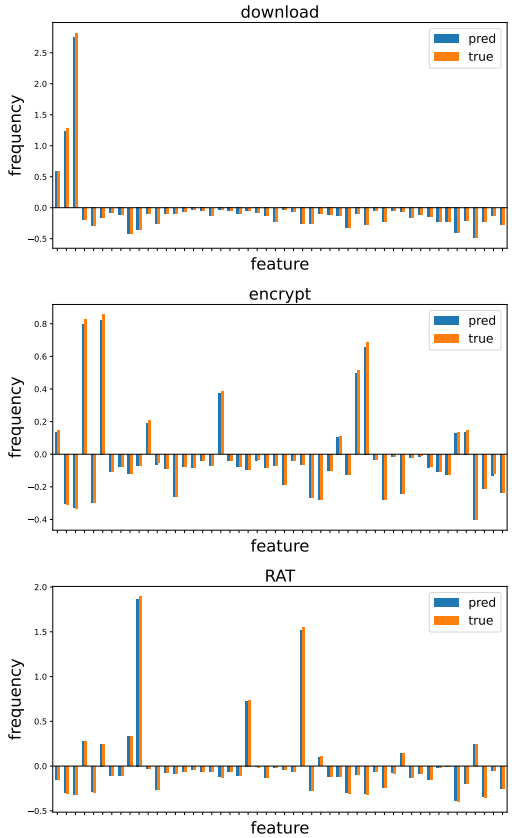

図 2　予測により正しく分類できた検体の特徴量（検体 (a)）
Fig. 2　Features of samples correctly classified by prediction (sample (a))

の平均二乗誤差が非常に大きくなった要因としては，一部の API が特に多く呼ばれていたからであると考えられる．

6.2　各目的ラベルに対する分類性能についての考察

表 3，表 4 から各目的ラベルによって分類精度（特に TPR）が大きく異なることが読み取れる．特に，他の目的ラベルに比べて encrypt と mining の TPR が低く，download，RAT，drop の TPR が比較的高い．以下では，そのような差が生じた原因について考察する．

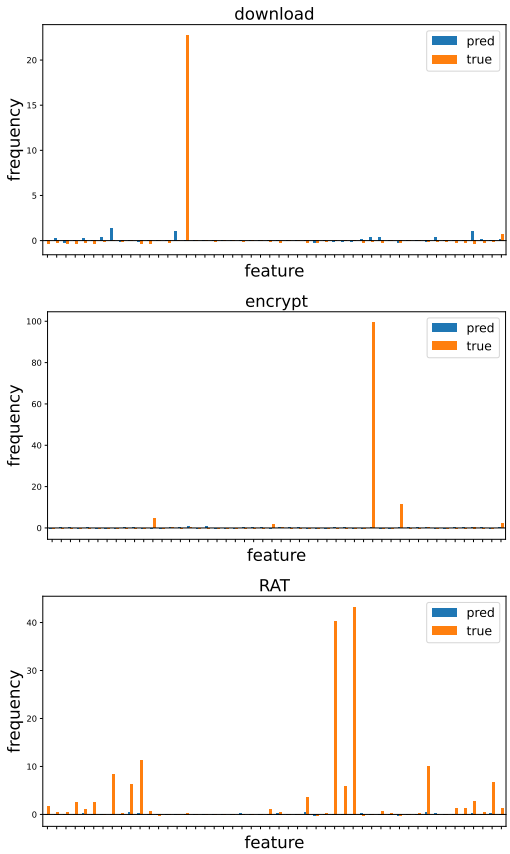

図 3　予測したが誤って分類された検体の特徴量（検体 (b)）
Fig. 3　Features of samples incorrectly classified by prediction (sample (b))

1 つ目に，今回の実験で使用した検体の目的ラベルの検体数が不均衡であることが考えられる．表 1 から，特に mining 検体は他の目的ラベルに比べて検体数が非常に少ないことが読み取れる．したがって，このような検体では分類器の学習時に検体の特徴を十分に学習できておらず，TPR が低下していることが考えられる．

2 つ目に，初期動作にマルウェアの目的を示す特徴が十分に含まれていなかったことが考えられる．ここで，encrypt の目的を持つ検体のうち，イベントログを用いた分類で誤分類された検体を抽出し，その分類後の目的ラベルの内訳を**表 8** に示した．表 8 から，encrypt の目的を持つ検体が，encrypt に分類されずに infostealer や RAT に分類されていることが読み取れる．例として，encrypt の目的のみを持つ 1 検体と，RAT の目的のみを持つ 1 検体のイベント頻度特徴量を抜き出してヒストグラム化したものを**図 4** に示した．図 4 の横軸はイベントログ特徴量，縦軸は特徴量の値となっており，青色の棒は encrypt 検体，赤色の棒は RAT 検体である．図 4 から，encrypt 検体に特徴量の一

表 8　誤分類された encrypt 検体の分類結果

Table 8　Classification results of misclassified encrypt samples

download	spread	infostealer	mining	RAT	drop
13	2	25	3	37	11

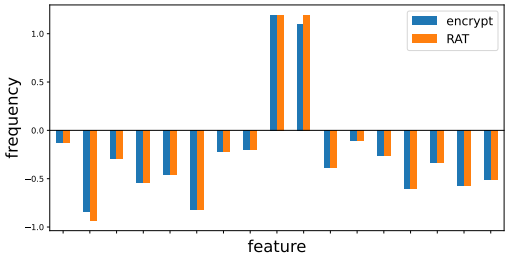

図 4　encrypt 検体と類似していた RAT 検体のイベントログ特徴量

Fig. 4　Event log features of RAT samples similar to encrypt samples

部がよく類似する RAT 検体が存在することがわかる．したがって，一部の encrypt 検体の初期動作には RAT などの他の検体の特徴が強く現れており，それらの検体の分類に失敗したため，TPR が低下したと考えられる．同様に，TPR が高い目的ラベルである download, RAT, drop などに関しては，初期動作にその目的を示す特徴が含まれていた場合が多いと考えられる．

7.　まとめと今後の課題

本稿では，短い期間のイベントログを用いた，より高精度なマルウェアの目的推定のため，イベントログの特徴量から API コールログの特徴量を予測し，それを用いて推定を行う手法を提案した．実験の結果，イベントログで目的推定を行う場合よりも TPR を 3.4%向上させることができ，API コールログを用いるときよりもログサイズを約 96%削減することができることが示された．

今後の課題として，予測性能のさらなる向上のための予測手法の検討が挙げられる．例えば，現時点で特徴量予測を高精度で行うことができない検体を考慮した，適切な特徴量抽出手法を検討する．

参考文献

[1] Jannatul Ferdous, et al. A review of state-of-the-art malware attack trends and defense mechanisms. *IEEE Access*, Vol. 11, pp. 121118–121141, 2023.

[2] Paul Cichonski, et al. Computer security incident handling guide, 2012.

[3] 佐々木良一. ネットワークセキュリティ. オーム社, 2014.

[4] Abdurrahman Pektaş and Tankut Acarman. Malware classification based on API calls and behaviour analysis. *IET Information Security*, Vol. 12, No. 2, pp. 107–117, 2018.

[5] Nitesh Kumar, et al. Malware classification using early stage behavioral analysis. In *2019 14th Asia Joint Conference on Information Security (AsiaJCIS)*, pp. 16–23. IEEE, 2019.

[6] Ferhat Ozgur Catak, et al. Deep learning based sequential model for malware analysis using windows exe api calls. *PeerJ computer science*, Vol. 6, p. e285, 2020.

[7] Ömer Aslan, et al. A new malware classification framework based on deep learning algorithms. *Ieee Access*, Vol. 9, pp. 87936–87951, 2021.

[8] Yang Xu and Zhuotai Chen. Family classification based on tree representations for malware. In *Proceedings of the 14th ACM SIGOPS Asia-Pacific Workshop on Systems*, pp. 65–71, 2023.

[9] Zhiguo Chen and Xuanyu Ren. An efficient boosting-based windows malware family classification system using multi-features fusion. *Applied Sciences*, Vol. 13, No. 6, p. 4060, 2023.

[10] 森優輝ほか. マルウェアによる感染活動の目的推定に向けた動的解析ログに基づく分析. コンピュータセキュリティシンポジウム 2018 論文集, Vol. 2018, No. 2, pp. 1186–1193, 2018.

[11] Naoto Kawaguchi and Kazumasa Omote. Malware function estimation using api in initial behavior. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. 100, No. 1, pp. 167–175, 2017.

[12] 児玉光平ほか. マルウェアの各種類似度に基づく機能判定とその効果. 信学技報, Vol. 120, No. 264, pp. 13–16, 2020.

[13] Mandiant. capa: Automatically Identify Malware Capabilities, 2020. `https://cloud.google.com/blog/topics/threat-intelligence/capa-automatically-identify-malware-capabilities/?hl=en` (2024/08/07 参照).

[14] Md Mazharul Islam, et al. Chimera: Autonomous planning and orchestration for malware deception. In *2021 IEEE Conference on Communications and Network Security (CNS)*, pp. 173–181. IEEE, 2021.

[15] Huaqi Sun, et al. Malware2att&ck: A sophisticated model for mapping malware to att&ck techniques. *Computers & Security*, Vol. 140, p. 103772, 2024.

[16] Ruitao Feng, et al. Unmasking the lurking: Malicious behavior detection for iot malware with multi-label classification. In *Proceedings of the 25th ACM SIGPLAN/SIGBED International Conference on Languages, Compilers, and Tools for Embedded Systems*, pp. 95–106, 2024.

[17] 寺田真敏ほか. 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〜. 研究報告情報基礎とアクセス技術 (IFAT), Vol. 2020, No. 8, pp. 1–6, 2020.

[18] 朝倉紗斗至ほか. 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 論文集, pp. 602–609, 2020.

[19] Takuya Akiba, et al. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[20] Eva Gibaja and Sebastián Ventura. A tutorial on multilabel learning. *ACM Computing Surveys (CSUR)*, Vol. 47, No. 3, pp. 1–38, 2015.

[21] Shantanu Godbole and Sunita Sarawagi. Discriminative methods for multi-labeled classification. In *Pacific-Asia conference on knowledge discovery and data mining*, pp. 22–30. Springer, 2004.