セクション情報を考慮したアンサンブル型マルウェア分類器の提案

SCIS 2023 2023 Symposium on
Cryptography and Information Security
Fukuoka, Japan, Jan. 24 – 27, 2023
The Institute of Electronics,
Information and Communication Engineers

セクション情報を考慮したアンサンブル型マルウェア分類器の提案
Ensemble Malware Classifier considering Section Information

竹内 廉 * 三橋 力麻 * 西垣 正勝 * 大木 哲史 *
Ren Takeuchi Rikima Mitsuhashi Masakatsu Nishigaki Tetsushi Ohki

あらまし サイバー攻撃者とセキュリティ解析者の戦争が日々激化している．最近のマルウェアは巧妙化が進み，簡単に入手・作成可能なツールの存在から亜種や新種へと多様化し続けている．これにより，セキュリティ解析者の負担が増加し，迅速な解析の妨げとなっている．多様化されたマルウェアの効率的な解析には，マルウェアファミリーの特定が重要であり，近年では低コストかつ汎用的に分類可能な深層学習ベースの手法が多く提案されている．その中でも，特徴量としてバイナリを画像で表現したマルウェア画像が多く用いられている．しかし，これまでの検討ではマルウェア分類に特化したモデルやアーキテクチャの提案がなされていない．著者らは過去にマルウェア画像の特徴調査を実施し，識別性が高いセクションの存在を確認した．そこで，全セクションを含む画像の分類モデルと識別性の高いセクションのみからなる画像の分類モデルを用意し，それらの予測結果を組み合わせることで，大局的特徴量と局所的特徴量の両者を考慮したアーキテクチャを提案する．BIG2015 データセットを用いた検証にて，ベースラインと比較して精度向上が見られ，提案手法の有意性を示した．

キーワード マルウェア分類，セクション情報，アンサンブル学習，畳み込みニューラルネットワーク

1 はじめに

サイバーセキュリティの分野において，サイバー攻撃者とセキュリティ解析者の戦争が日々激化している．セキュリティ解析者は，マルウェアの構造や挙動を解析することで，その検知方法に加え，発生し得る被害やマルウェアの攻撃能力，攻撃者の意図などを把握し，マルウェアへの対策を講じる．しかし，最近のマルウェアは新たな難読化技術や脆弱性をついた攻撃を利用することで巧妙化しており，なおかつマルウェアを簡単に入手および作成可能なツール [1, 2] が存在し，誰でも新たなマルウェアを生成できるため，亜種や新種といったマルウェアが増加し続けている．これにより，マルウェアの多様化が進み，セキュリティ解析者は手作業による解析の負担を強いられ，迅速なマルウェア解析実行の妨げとなっている．実際に，AV-TEST 研究所は過去 5 年間で毎年 1 億前後 [3]，McAfee 社は 2021 年の第 1 四半期で約 8760 万 [4] といった新しいマルウェアを発見している．これらの報告からも，攻撃者が既存のマルウェアを利用するだけでなく，標的とする環境に効果的なマルウェア（亜種や新種）へと進化および適応させ続けていることがわかる．

多様化された大量のマルウェアを効率的に解析するためには，マルウェアファミリーを特定することが重要である．ここで，マルウェアファミリーとは，オリジナルのマルウェアとそれを一部改変して作られた亜種マルウェアをひとまとめにしたものである．攻撃者は新たなマルウェアの作成コストを抑えるために，多くの亜種マルウェアを作成する傾向にあり，新たに出現したマルウェアの多くは亜種マルウェアであることが示唆されている [5]．そのため，マルウェアファミリーに基づいて分類することは，過去の事例による知識と経験を活かし，解析対象のマルウェアが持つ特徴を予想した上で解析に臨めるため，セキュリティ解析者にとって有益である．

マルウェアファミリーの分類は，これまでパターンマッチングによる手法や機械学習ベースの手法が用いられてきた．パターンマッチングによる手法は，マルウェア解析者が事前に定義した特定のコードやハッシュ値などの特徴（データベース）と比較し，最も近いものに分類する．機械学習ベースの手法では，パターンマッチング方式を上回る高速化，さらには分類用データベースを充実させることで分類精度の向上が見込める．しかし，手動によるデータベース作成が必要なため，多くのコストがかかり，人的エラーが発生する恐れもある．また，同じマルウェアファミリーに属するマルウェアであっても常に難読化によってコードが改変されるため，難読化されたマルウェアへの対応が困難となる [6]．

* 静岡大学大学院総合科学技術研究科, 静岡県浜松市中区城北 3 丁目 5-1, Shizuoka University, 3-5-1 Jo-hoku, Naka-ku, Hamamatsu City, Shizuoka, Japan

これらの問題を克服するため，マルウェアを低コストかつ汎用的に分類可能な深層学習ベースの手法が近年多く提案されている [7, 8]．深層学習は多層構造のニューラルネットワークを用いることで，大量にラベル付けされたデータから直接特徴を学習することができ，手作業による特徴抽出を必要としない．本研究では，特徴量としてマルウェア画像を利用した手法 [9, 10, 11, 12, 13] に着目する．バイナリを画像で表現することで，マルウェアファミリーごとの特徴が視覚的に捉えられ，それらの特徴が同じファミリーに属するマルウェア同士で類似することから高精度な分類が可能である．

しかし，これまでの先行研究では，マルウェアファミリー分類に特化したモデルやアーキテクチャの提案がなされていない．分類精度の向上を目的として，視覚化手法に対するアプローチをとる研究 [12, 13] や画像分類で成功を収めているモデルをそのまま適用する研究 [10, 11] が多い．マルウェアが更なる多様化を見せ，画像による分類がより困難になることが予想されるため，マルウェアファミリー分類に特化したモデルの構築が求められる．

そこで，我々はマルウェア画像の特徴調査を実施し，ファミリー内の識別性が高いセクションの存在を確認した [14]．さらに，特定のセクションのみを抽出した画像（以降，単体画像）で学習したモデルを複数組み合わせることで，すべてのセクションが含まれるマルウェア画像（以降，元画像）で学習したモデルの分類精度を上回る可能性があると示唆している．これらをふまえ，既存手法である元画像で学習した CNN モデル（以降，元画像モデル）と識別性の高いセクションの単体画像で学習した複数のモデルを用意し，それらの予測結果を組み合わせるアンサンブル学習を応用することで，元画像を用いた大局的特徴量に加えて，各セクションの局所的特徴量を考慮した，セクションアンサンブルアーキテクチャを提案する．

本提案手法では，バイナリ情報とセクション位置情報が必要であるため，これらが既に揃っている BIG2015 データセット [15] にて分類精度を検証した．元画像モデルをベースラインとして比較した結果，0.28%の精度向上が見られた．

本研究の貢献は，次のようにまとめられる．

- 元画像モデルと識別性の高いセクションの単体画像で学習したモデルの予測結果を組み合わせるセクションアンサンブルアーキテクチャを提案する．
- BIG2015 データセットを用いた検証にて，元画像モデルと比較して提案手法の分類精度が向上することを示した．

2 関連研究

2.1 画像ベースマルウェアファミリー分類

マルウェアファミリーの分類は，多様化されたマルウェアを効果的に解析するための最初のステップであり，マルウェアを視覚化した上で分類する手法がこれまで多く提案されてきた．

Nataraj ら [9] は，マルウェアバイナリをグレースケール画像として視覚化し，ユークリッド距離による k 近傍法にてマルウェアを分類する手法を提案した．同じファミリーのマルウェア画像はレイアウトやテクスチャが類似するとされており，この類似性から逆アセンブルやコード実行が不要で，標準的な画像特徴量を用いた手法でも十分に分類できる．この手法の問題点として，浅い機械学習技術の使用から，サンプル数の増加に対応できない点，手動での特徴抽出が必要である点が挙げられる．

これらの問題点に対処するため，グレースケール画像と深層学習を組み合わせた，より堅牢で汎用的な手法が提案されている．Kalash ら [10] と Rezende ら [11] は，画像分類において成果をあげている CNN モデル（VGG16, ResNet50）を導入することで，高い分類精度を達成した．

また，バイナリとは異なる特徴量を視覚化する研究もなされている．Ni ら [12] は，Opcode を特徴量として，SimHash というハッシュアルゴリズムを使ってグレースケール画像を生成する手法を提案した．Ren ら [13] は，フラクタル曲線を使ってバイト列の unigram を視覚化する手法，bigram とその統計情報をピクセルの座標および輝度とみなして視覚化する手法の 2 種類を提案した．

これらの先行研究は，分類精度の向上を目的として，視覚化手法に対するアプローチや画像分類で成功を収めたモデルをそのまま利用するアプローチが多く，マルウェアファミリー分類に特化したモデルとはいえない．

2.2 セクション情報の考慮

著者らは文献 [14] において，PE ファイルを対象としたマルウェア画像の特徴調査を実施し，マルウェア画像の識別性とセクション情報との関係性を明らかにした．文献 [14] では，まず，ソースコードに難読化を施した上でコンパイルした実行ファイルのバイナリを用意し，難読化前のバイナリと難読化後のバイナリを視覚化した．その結果，複数の特徴ある領域（セクション）に分割されていること，難読化方法によって異なる変動が生じ，特に.text セクションの変動が大きいことを確認した．そして，セクションごとに難読化前後のサイズ増加量を調査し，.text セクションにおけるサイズ変動が最も大きいことを確認した．さらに，難読化によって，セクションに依存した変動パターンが存在し，マルウェアファミリー分類への影響度がセクションごとに異なると仮定し

て，セクションごとに生成した単体画像を用いてモデルの学習および分類精度の比較を行った．ここで，単体画像とは，.text，.data，.rdata，.idata 等の特定のセクション領域のみを切り出したマルウェア画像である．その結果，.text，.data，.rdata の 3 つのセクションは，単体画像でも十分な分類精度を達成可能であり，ファミリー内の識別性が高いセクションであることを確認した．

類似の研究として，Xiao ら [16] は，セクション分布情報を強調する色付きのラベルボックスをマルウェア画像に導入することで，元々のグレースケール画像で学習したモデルと比較して分類精度が向上した．

これらの結果をふまえれば，マルウェアのセクション情報が分類に貢献すること，また，これを考慮したモデルとすることで更なる精度向上が見込めると考えられる．

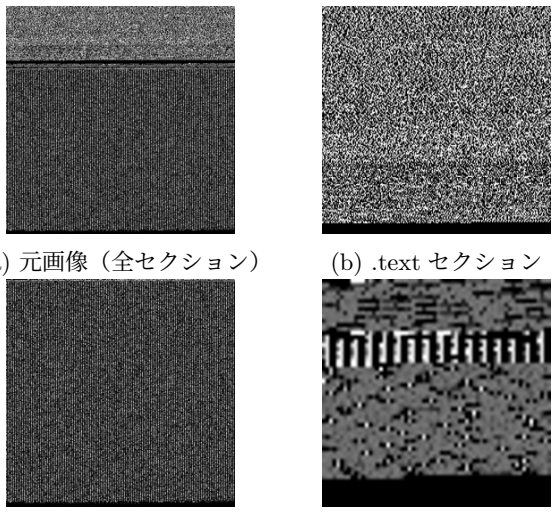
2.3 アンサンブル学習の応用

アンサンブル学習とは，個々に学習した複数のモデルを融合させることで，分類失敗の主な原因であるノイズやバイアス，バリアンスを最小化して，モデルの汎化性能向上をはかる手法である．

アンサンブル学習をマルウェアファミリー分類に応用した手法がいくつか提案されている．Yan ら [17] は，グレースケール画像で学習した CNN の予測結果，Opcode シーケンスで学習した LSTM の予測結果，メタデータ特徴量の 3 種を統合して，最終的な分類結果を出すロジスティック回帰分類器を作成する手法を提案した．Narayanan ら [18] は，CNN4 種と LSTM から抽出した特徴量を組み合わせて，ロジスティック回帰や SVM を使用した分類を行うアンサンブルアプローチを提案した．いずれの手法も BIG2015 データセットに対して 99%を超えた分類精度を達成しており，アンサンブル学習の応用が効果的であることが示されている．

3 提案手法

本稿では，既存手法である元画像モデルと識別性の高いセクションの単体画像で学習した複数モデルの予測結果を，アンサンブル学習にて組み合わせる手法（以下，セクションアンサンブルアーキテクチャ）を提案する．これにより，元画像から抽出される大局的特徴量と各セクションの単体画像から抽出される局所的特徴量を考慮したマルウェア分類を可能とする．ここで，本稿における元画像をすべてのセクションを含めたマルウェア画像，単体画像を特定のセクション領域のみを切り出したマルウェア画像とそれぞれ定義し，生成する元画像および単体画像の例を図 1 に示す．なお，本提案手法では，文献 [14] で得られた成果に基づき，識別性の高いセクションとして，.text，.data，.rdata の 3 セクションを選択した．



(a) 元画像（全セクション） (b) .text セクション

(c) .data セクション (d) .rdata セクション

図 1: 元画像および対象セクションごとの単体画像

提案手法の概要図を図 2 に示す．提案手法は，画像生成フェーズ，特徴抽出フェーズ，アンサンブルフェーズの 3 つにより構成される．各フェーズの詳細を 3.1 節から 3.3 節にて説明する．

3.1 画像生成フェーズ

画像生成フェーズでは，マルウェアのバイナリ情報とセクションのアドレス情報を用いて元画像および対象セクションの単体画像を生成する．本稿では，Nataraj らの変換手法 [9] に従って，バイナリから画像に変換する．画像生成フェーズの主な流れは以下の通りである．

1. マルウェアのプログラム構造を解析して，各セクションの位置情報（開始アドレスと終了アドレス）を取得する．
2. バイナリを 1Byte ずつ 1 次元配列に読み込む．このとき，各配列の要素は 10 進数で 0〜255 の値となる．
3. 1 次元配列から，対象セクションの開始アドレスから終了アドレスまでの要素を抽出する．
4. バイナリのサイズに応じて画像の幅を設定し，1 次元配列から 2 次元配列に変換する．サイズに対応した画像の幅は Nataraj らの研究で定義されたものをそのまま使用する．
5. 2 次元配列の各要素を画素値としてグレースケール画像を生成する．

3.2 特徴抽出フェーズ

特徴抽出フェーズでは，生成した元画像および対象セクションの単体画像を用いて，CNN による特徴埋め込

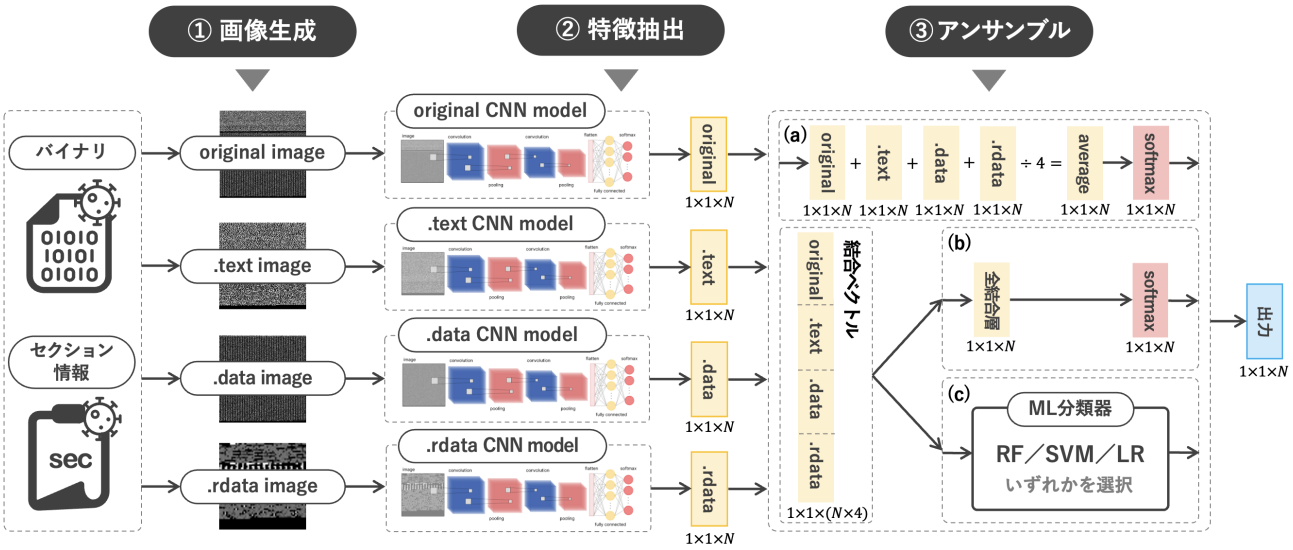


図 2: 提案手法（セクションアンサンブルアーキテクチャ）の概要図

み（高次元ベクトルの低次元空間への変換）を学習する．特徴抽出フェーズの主な流れは以下の通りである．

1. CNN を用いて，元画像または単体画像を入力とした，マルウェアファミリーの多クラス分類タスクを学習する．
2. 学習済みの CNN から softmax 関数を取り除くことで，softmax 関数の 1 つ手前の全結合層を出力とする特徴抽出器を用意する．

3.3 アンサンブルフェーズ

アンサンブルフェーズでは，元画像および単体画像から抽出した特徴量をもとに，既存のアンサンブル手法を応用して最終的なファミリーを推定する．本稿では，平均型，特徴統合型，ML（Machine Learning）分類型の 3 種類を用意した．

(a) 平均型 特徴量の各要素の総和に対して，モデル数で割った平均を出力特徴量として学習する．

(b) 特徴統合型 特徴量を結合したベクトルを入力とし，ファミリー数 N を出力要素数とする全結合層を持つ多クラス分類器を学習する．

(c) ML 分類型（スタッキング） 特徴量を結合したベクトルを入力とし，ML 分類器で判定する．本稿では，ML 分類器が用いる機械学習アルゴリズムとしてランダムフォレスト（以降，RF），サポートベクターマシン（以降，SVM），ロジスティック回帰（以降，LR）のいずれかを選択する．

4 実験・検証

本検証では，提案手法であるセクションアンサンブルアーキテクチャがマルウェアファミリー分類に効果的かどうかを検証する．

4.1 検証方法

提案手法の有効性を示すため，元画像を用いる手法（ベースライン），およびセクション単体画像を用いる手法の各組み合わせを検証することで，元画像から抽出される大局的特徴量と単体画像から抽出される局所的特徴量の組み合わせによる分類精度への影響を確認する．本実験で検証する 3 つのシナリオを次にまとめる．

- すべてのセクションを含む，元画像を使って CNN モデルを学習する既存手法（ベースライン）
- 対象セクションの単体画像で学習した CNN3 種をアンサンブル学習で組み合わせる提案手法（ベースラインの組み込みなし）
- 対象セクションの単体画像で学習した CNN3 種とベースラインの計 4 種をアンサンブル学習で組み合わせる提案手法（ベースラインの組み込みあり）

4.2 検証に使用したデータセット

4.2.1 使用するデータセットの概要

本検証では，検証用データセットとして BIG2015 データセット [15] を用いた．BIG2015 データセットは，Microsoft が Kaggle にて開催した「Microsoft Malware Classification Challenge」コンテストで公開されたもので，21,741 個のマルウェアサンプルからなるデータセットである．このデータセットは，学習用の 10,868 サンプル

表 1: BIG2015 データセットの構成

ファミリー名	学習用データの サンプル数	3 セクションを 含むサンプル数
Ramnit	1541	1134
Lollipop	2478	2401
Kelihos_ver3	2942	2934
Vundo	475	436
Simda	42	34
Tracur	751	132
Kelihos_ver1	398	358
Obfuscator.ACY	1228	317
Gatak	1013	917
合計	10868	**8663**

と，テスト用の 10,873 サンプルの 2 つに分かれており，各マルウェアサンプルは，9 種類のマルウェアファミリーのいずれかに属している．本検証では，各サンプルが属するマルウェアファミリー情報（ラベル）が付与された学習用データのみを利用する．

各マルウェアサンプルは，PE ヘッダーを削除した 16 進バイナリとアドレスを含む.bytes ファイル，IDA で抽出された逆アセンブルコードを含む.asm ファイルの 2 つで構成される．通常，マルウェア画像の生成には.bytes ファイルのみを用いるが，本検証ではセクション単体画像を用意するために，.asm ファイルも用いる．

4.2.2 セクションの位置情報取得

BIG2015 データセットにおけるセクションの位置情報取得には，データセットに含まれている.asm ファイルを利用する．.asm ファイルにはバイト列やアドレスだけでなく，Opcode やセクション名などの情報も含まれている．その中でも，セクションの位置情報取得にはセクション名とアドレスを利用する．方法としては，.asm ファイルを 1 行目から読み込み，セクション名が切り替わる境界部分を検知することで，各セクションの開始アドレスおよび終了アドレスを求める．

4.2.3 検証用データセットの構成

BIG2015 データセットの元構成と本検証で使用するサンプルの構成を表 1 に示す．4.2.2 項にあるように，サンプルに含まれるセクションを.asm ファイルのセクション名で判断しているため，セクション情報の暗号化，攻撃者による手動での書き換え等により，対象セクションが含まれていないと判断されるサンプルが多く存在する．本提案手法では，対象セクションの単体画像を必要とするため，本検証においては対象セクションの 3 つをすべて含むサンプルのみを抽出して使用する．

表 2: 本検証における混同行列

		実測	
		Positive	Negative
予測	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

4.3 検証に使用する CNN モデル

本検証では，マルウェア画像を入力とする特徴抽出器に利用する CNN モデルとして，VGG16[19] と ResNet50[20] を使用する．まず，PyTorch の torchvision に用意されている事前学習済みの VGG16 および ResNet50 を使用し，重みを凍結せずに学習データを用いたファインチューニングを行うことでマルウェアファミリー分類器を作成する．そして，作成した分類器から softmax 関数を取り除いたモデルを特徴抽出器として使用する．なお，検証に使用する BIG2015 データセットは 9 クラスであるため，全結合層の最終層を $[1 \times 1 \times 9]$ に変更している．

4.4 検証環境

本検証には，AMD Ryzen9 3950X プロセッサ，64GB メモリ，GPU として GeForceRTX2080Ti（11GB GDDR メモリ）を 1 基搭載した PC を用いた．なお，OS は Ubuntu 20.04.1 LTS，OS のカーネルは Linux 5.4.0-56-generic を用いた．この検証機器で使用するソフトウェアは Python バージョン 3.6.10，PyTorch バージョン 1.8.0，scikit-learn バージョン 0.23.2 を用いた．

4.4.1 モデルの学習と検証方法における設定

VGG16 の学習は，Kalash らの研究 [10] を可能な限り再現した．損失関数としてクロスエントロピーロス，オプティマイザーとして確率的勾配降下法（SGD）を使用し，バッチサイズ 6 で 25 エポック学習させた．SGD のパラメータは，学習率 0.001，モメンタム 0.9 であり，学習率に関しては，20 エポック毎に 10 倍ずつ減少させている．また，エポック毎に学習データをランダムにシャッフルしている．なお，ResNet50 に関しても同様の条件下で検証を行っている．

アンサンブルフェーズにて使用する 3 種類の ML 分類器は，Python の機械学習ライブラリである scikit-learn で用意されているものを利用する．ML 分類器のハイパーパラメータチューニングには，自動最適化フレームワークである Optuna を用いており，使用するアルゴリズムを TPE，試行回数を 100 回にそれぞれ設定した上で探索を行っている．

検証には K 分割交差検証を用いた．これはデータを

表 3: ベースラインと提案手法の分類精度

ベースライン or 提案手法	モデルの種類	VGG16			ResNet50		
		Accuracy	Precision	Recall	Accuracy	Precision	Recall
ベースライン（元画像モデル）		99.18%	99.19%	99.18%	99.15%	99.17%	99.15%
提案手法（ベースラインの組み込みなし）	平均型	**99.18%**	**99.19%**	**99.18%**	**99.21%**	**99.23%**	**99.21%**
	特徴統合型	98.73%	98.73%	98.73%	98.85%	98.85%	98.85%
	ML 分類型-RF	99.09%	99.10%	99.09%	99.09%	99.11%	99.09%
	ML 分類型-SVM	99.06%	99.07%	99.06%	99.08%	99.09%	99.08%
	ML 分類型-LR	99.15%	99.15%	99.15%	99.12%	99.14%	99.12%
提案手法（ベースラインの組み込みあり）	平均型	99.34%	99.35%	99.34%	**99.43%**	**99.45%**	**99.43%**
	特徴統合型	99.19%	99.19%	99.19%	99.20%	99.20%	99.20%
	ML 分類型-RF	99.32%	99.33%	99.32%	99.24%	99.25%	99.24%
	ML 分類型-SVM	99.23%	99.29%	99.23%	99.32%	99.34%	99.32%
	ML 分類型-LR	**99.39%**	**99.40%**	**99.39%**	99.38%	99.39%	99.38%

K 個に分割してその内の 1 つをテストデータに，残りの $K-1$ 個を学習データとしてモデルの学習および評価を行い，K 回分の精度を平均する手法である．今回は $K=5$ に設定し，学習セット，検証セット，テストセットの 3 つのサブセットに分割する割合は (学習, 検証, テスト) = (60%, 20%, 20%) とした．また，各マルウェアの画像サイズが異なるため，すべての検証において (縦, 横) = (224, 224) に統一した．

4.4.2 評価指標

本検証では，精度を評価する際の指標として，正確度（Accuracy），適合率（Precision），再現率（Recall）の 3 つを採用した．正確度は全予測の正答率，適合率はモデルが Positive と判定したサンプルのうち，実際に正解ラベルが Positive である割合，再現率は正解が Positive なサンプルのうち，正しく Positive と判定した割合である．マルウェアファミリー分類は多クラス分類タスクであるため，ある 1 つのクラスを Positive，それ以外を Negative として各指標の値を算出した後，それらの平均を取ることで全体的な値を算出する．N クラス（クラス 1〜クラス N）の分類タスクを想定すると，各評価指標の計算式は以下のように表せる．なお，本研究における混同行列は表 2 に示し，クラス n における TP, TN, FP, FN をそれぞれ TP_n, TN_n, FP_n, FN_n，各クラス n の値を平均した全体的な評価指標の値をそれぞれ $Accuracy$，$Precision$，$Recall$ とする．

$$Accuracy = \frac{1}{N}\sum_{n=1}^{N}\frac{TP_n + TN_n}{TP_n + FP_n + FN_n + TN_n} \quad (1)$$

$$Precision = \frac{1}{N}\sum_{n=1}^{N}\frac{TP_n}{TP_n + FP_n} \quad (2)$$

$$Recall = \frac{1}{N}\sum_{n=1}^{N}\frac{TP_n}{TP_n + FN_n} \quad (3)$$

4.5 検証結果

4.1 節に従い，BIG2015 データセットを用いた検証実験を行った．ベースラインと提案手法におけるそれぞれの分類精度を表 3 に示す．なお，提案手法におけるモデルは，CNN モデル 2 種（VGG16，ResNet50）とアンサンブル手法 5 種（平均型，特徴統合型，ML 分類型-RF，ML 分類型-SVM，ML 分類型-LR）を組み合わせた 10 種類を用意して検証を行った．

4.5.1 ベースラインの分類精度

まず，提案手法との比較対象として，ベースラインとなる元画像モデルの分類精度を検証した．なお，表 3 の「ベースライン（元画像モデル）」に該当する．

VGG16 では 99.18%，ResNet50 では 99.15%の Accuracy を達成した．これは，先行研究にて示された分類精度と概ね同じ値となっている．

4.5.2 提案手法の分類精度（アンサンブル学習にベースラインを組み込まない場合）

次に，各セクションの単体画像で学習した CNN3 種をアンサンブル手法にて組み合わせた際の分類精度を検証した．なお，表 3 の「提案手法（ベースラインの組み込みなし）」に該当する．

VGG16 と ResNet50 それぞれ 5 種のモデルにおいて，どちらも平均型が最も高い分類精度となり，VGG16 は 99.18%，ResNet50 は 99.21%の Accuracy を達成した．また，ベースラインと比較して ResNet50 は 0.06%の精度向上が見られたが，VGG16 は同じ精度となり，平均型以外の 4 種は VGG16 と ResNet50 どちらもベースラインより低い分類精度となった．

表 4: 先行研究の分類精度

モダリティ	分類手法	年	評価方法	Accuracy
単一 (バイナリ)	MCSC(SimHash+CNN)[12]	2018	記載なし	98.86%
	MalCVS(CoLab 画像+VGG16+SVM)[16]	2021	10 回交差検証	98.94%
	M-CNN(VGG16)[10]	2018	記載なし	98.99%
	Ensemble(CNN4 種)+SVM[18]	2020	ホールドアウト検証	99.4%
	提案手法 (ResNet50 の平均型)	**2023**	**5 回交差検証**	**99.43%**
複数 (バイナリ+Opcode)	MalNet(CNN+LSTM+LR)[17]	2018	記載なし	99.36%
	Ensemble(CNN4 種+LSTM)+SVM[18]	2020	ホールドアウト検証	**99.8%**

4.5.3 提案手法の分類精度（アンサンブル学習にベースラインを組み込む場合）

最後に，各セクションの単体画像で学習した CNN3 種とベースラインの計 4 種をアンサンブル手法にて組み合わせた際の分類精度を検証した．なお，表 3 の「提案手法（ベースラインの組み込みあり）」に該当する．

VGG16 では，ML 分類型-LR が最も高い分類精度となり，99.39%を達成した．ResNet50 では，平均型が最も高い分類精度となり，99.43%を達成した．また，ベースラインを組み込んだ提案手法のモデル 10 種すべてにおいて，ベースラインの分類精度を上回った．

5 議論

5.1 アンサンブル学習におけるベースラインの組み込み

表 3 より，セクション単体画像で学習した複数のモデルとベースラインの両方をアンサンブル学習に組み込むことで，アンサンブル手法に依存せずベースラインの分類精度を上回っている．一方，ベースラインをアンサンブル学習に組み込まないモデルでは，大半のアンサンブル手法においてベースラインの分類精度を下回っている．この結果から，本提案手法が仮定した通り，元画像から抽出できる特徴量と単体画像から抽出できる特徴量の違いを考慮し，両者を組み合わせることで精度向上が達成可能であることが確認された．具体的には，CNN モデルを使った特徴抽出器によって，元画像からセクションの並び方に応じた，より大局的な特徴量が抽出でき，単体画像から各セクションのテクスチャに応じた，より局所的な特徴量を抽出できる．この両者の特徴量を考慮することで，これまで元画像においてセクションのレイアウトや全体のテクスチャが似ているファミリー間で誤分類が起きていたサンプルを正しく分類できたと推定できる．

5.2 先行研究との分類精度比較

本提案手法とこれまでの先行研究で示されている分類精度の比較を行う．BIG2015 データセットを用いて検証している先行研究の分類精度を表 4 に示す．提案手法の中では，ResNet50 の平均型が 99.43%を達成しており，バイナリのみを特徴量として使用している研究の中では，最も高い分類精度を誇っている．また，バイナリと Opcode の両者を使用している研究の中では，99.8%という非常に分類精度の高い研究があり，本提案手法では超えることができなかった．しかし，この研究はホールドアウト検証による評価をしており，データの分割方法によっては精度評価に影響を及ぼす可能性がある．そのため，該当研究を可能な限り再現し，交差検証を用いた評価を行った上で精度比較を行うべきだと考える．

なお，本検証では対象セクションをすべて含むサンプルのみを抽出しているため，先行研究との整合性がとれていない．全データにおける分類精度が現状の精度より低下する恐れもあるため，対象セクションが存在しない場合にも対応できるような仕組みを本提案手法に取り入れて，厳密な精度比較ができるようにする必要がある．

6 おわりに

本稿では，すべてのセクションを含む画像（元画像）で学習したモデルと識別性の高いセクションのみを抽出した画像（単体画像）で学習した複数のモデルを用意し，それらの予測結果をアンサンブル学習にて組み合わせるマルウェアファミリー分類モデルを提案した．本稿での検証にて，本提案手法とベースラインを比較した結果，最大で 0.28%の精度向上が見られた．この結果から，元画像モデルと単体画像で学習したモデルの両者をアンサンブル学習に組み込むことで，マルウェアバイナリの大局的な特徴量と局所的な特徴量のどちらも考慮する形となり，分類精度の向上に繋がったと考えられる．

今後は，本提案手法の有意性を確かなものにするために，より広範なデータセットやモデルを利用した検証，対象セクションを増減させた検証等を進めるとともに，対象セクションが存在しないサンプルにも対応したアーキテクチャへと拡張し，より正確な先行研究との精度比較を行う．また，各セクションの単体画像において，最適な CNN モデルがそれぞれ異なる可能性が考えられる

ため，それらの探索も進めていく必要がある．さらに，多様化が進む現実世界を想定し，過去のマルウェアを学習データ，未来のマルウェアを評価データとするデータセットを構築して本提案手法が対応可能かどうか検証を行う．そして，これらの検証結果を元に，更なるマルウェアファミリー分類に特化したモデルを検討し，迅速かつ確実なマルウェア対策の実現へと繋げていきたい．

参考文献

[1] McAfee Labs. Meet ‘tox’: Ransomware for the rest of us, 2015. https://www.mcafee.com/blogs/other-blogs/mcafee-labs/meet-tox-ransomware-for-the-rest-of-us/,Accessed:2022/12/01.

[2] BlackBerry. Threat spotlight: Eternity project maas goes on and on, 2022. https://blogs.blackberry.com/en/2022/06/threat-spotlight-eternity-project-maas-goes-on-and-on,Accessed:2022/12/01.

[3] AV-TEST. Malware statistics & trends report, 2022. https://www.av-test.org/en/statistics/malware/,Accessed:2022/11/17.

[4] McAfee. 2021 年 6 月 mcafee labs 脅威レポート, 2021. https://www.mcafee.com/enterprise/ja-jp/assets/reports/rp-threats-jun-2021.pdf,Accessed:2022/11/17.

[5] Lakshmanan Nataraj, Shanmugavadivel Karthikeyan, and B. S. Manjunath. Sattva: Sparsity inspired classification of malware variants. *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, 2015.

[6] Royi Ronen, Marian Radu, Corina Feuerstein, Elad Yom-Tov, and Mansour Ahmadi. Microsoft malware classification challenge. *ArXiv*, Vol. abs/1802.10135, , 2018.

[7] Adel Abusitta, Miles Q. Li, and Benjamin C. M. Fung. Malware classification and composition analysis: A survey of recent developments. *J. Inf. Secur. Appl.*, Vol. 59, p. 102828, 2021.

[8] Rupali Komatwar and Manesh Kokare. Retracted article: A survey on malware detection and classification. *Journal of Applied Security Research*, Vol. 16, pp. 390–420, 2020.

[9] Lakshmanan Nataraj, Shanmugavadivel Karthikeyan, Grégoire Jacob, and B. S. Manjunath. Malware images: visualization and automatic classification. In *VizSec ’11*, p. 1–7, 2011.

[10] Mahmoud Kalash, Mrigank Rochan, Noman Mohammed, Neil D. B. Bruce, Yang Wang, and Farkhund Iqbal. Malware classification with deep convolutional neural networks. *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, 2018.

[11] Edmar Rezende, Guilherme Ruppert, Tiago Carvalho, Fabio Ramos, and Paulo de Geus. Malicious software classification using transfer learning of resnet-50 deep neural network. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1011–1014, 2017.

[12] Sang Ni, Quan Qian, and Rui Zhang. Malware identification using visualization images and deep learning. *Comput. Secur.*, Vol. 77, pp. 871–885, 2018.

[13] Zhuojun Ren, Guang Chen, and Wenke Lu. Malware visualization methods based on deep convolution neural networks. *Multimedia Tools and Applications*, Vol. 79, pp. 10975–10993, 2019.

[14] 竹内廉, 西垣正勝, 大木哲史. 画像ベースマルウェア分類器に対するセクション情報が与える影響. コンピュータセキュリティ研究会 2022(CSEC2022-05), pp. 1–8, 2022.

[15] Kaggle. Microsoft malware classification challenge, 2015. https://www.kaggle.com/c/malware-classification/,Accessed:2022/11/17.

[16] Mao Xiao, Chun Guo, Guowei Shen, Yunhe Cui, and Chaohui Jiang. Image-based malware classification using section distribution information. *Comput. Secur.*, Vol. 110, p. 102420, 2021.

[17] Jinpei Yan, Yong Qi, and Qifan Rao. Detecting malware with an ensemble method based on deep neural network. *Security and Communication Networks*, Vol. 2018, pp. 1–16, 2018.

[18] Barath Narayanan Narayanan and Venkata Salini Priyamvada Davuluru. Ensemble malware classification system using deep neural networks. *Electronics*, Vol. 9, p. 721, 2020.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.