情報処理学会論文誌 Vol.62 No.12 1879–1892 (Dec. 2021)

データ欠損を起こしたマルウェアの機械学習による名称同定および悪性判定

小久保 博崇[1,a)] 江田 智尊[1] 大山 恵弘[2]

受付日 2021年2月25日, 採録日 2021年9月9日

概要：サイバー攻撃の被害状況を特定するにあたって，攻撃に使われたマルウェアを知ることは重要である．しかし，調査妨害を目的とした自己消去を行うマルウェアも存在する．消去されたファイルはOS上から参照することができなくなるが，環境によってはストレージ上に残存している．ただし，消去されたファイルはストレージの使用とともにデータが欠損していくため，フォレンジック技術を使用しても完全な形でマルウェアを復元できないことがある．本研究では，データの欠損を起こしたマルウェア（欠損マルウェア）に着目する．著者らは過去に，欠損マルウェアをアンチウイルスによって同定することは難しいという実験結果を示した．本研究では，欠損マルウェアに対する機械学習技術を用いたマルウェア名同定を試みた．欠損マルウェアはデータの欠損を起こしており，ヘッダ情報や挙動解析情報などの有用な特徴が得られないため，同定には画像特徴量を使用した．その結果，アンチウイルスによる同定に致命的な悪影響を与えるファイル先頭の欠損があったとしても，欠損前のマルウェア検体を学習に用いれば約97%の精度，欠損前のマルウェア検体と同種の検体を学習に用いれば約48%の精度で，マルウェア名の同定が可能であった．また，良性ソフトウェアをデータセットに含んだ実環境に近い環境でも実験を行い，その場合においてもそれぞれ約85.7%，約38.5%の精度でマルウェア名の同定が可能であることが分かった．

キーワード：マルウェア，機械学習，データ欠損，マルウェア同定，フォレンジック

Identification of Data Corrupted Malware Using Machine Learning

Hirotaka Kokubo[1,a)] Satoru Koda[1] Yoshihiro Oyama[2]

Received: February 25, 2021, Accepted: September 9, 2021

Abstract: In order to know the damage situation of cyber attack, it is important to know what kind of malware was used for the attack. Some malware erases itself to prevent investigation. Erased malware can be recovered by digital forensics technology, but data loss can occur. In this paper, we identify malware names from corrupted malware binary by using machine learning. We use image feature values to perform identification because header information and behavior analysis information cannot be used due to data loss. As a result, it was possible to identify the malware name with an accuracy of about 97% when the same malware before the data loss was used for training, and with an accuracy of about 48% when the same kind of malware as that before the data loss was used for training. Even when benign software was included in the dataset, the malware names could be identified with an accuracy of about 85.7% and 38.5%, respectively.

Keywords: malware, machine learning, data loss, malware identification, digital forensics

1. はじめに

組織がサイバー攻撃を受けたとき，攻撃に使われたマルウェアを知ることは，被害状況を特定するうえで重要である．攻撃に使われたマルウェアが同定できれば，公開情報を利用することにより，マルウェアの除去方法だけでなく

[1] 富士通株式会社
Fujitsu Limited, Kawasaki, Kanagawa 211–8588, Japan
[2] 筑波大学
University of Tsukuba, Tsukuba, Ibaraki, 305–8577, Japan
[a)] kokubo.hirotaka@fujitsu.com

情報流出の可能性や組織内感染拡大の状況把握など，インシデントレスポンスに役立つ情報を得ることができる．

どのようなマルウェアが使われたかを知る手段の 1 つは，マルウェアに感染したマシンのストレージを調査することである．マルウェアがファイルとしてコンピュータ上に残存していれば，アンチウイルスソフトによるスキャンや専門家の調査により検体を取得し，どのマルウェアにより被害を受けたかを知ることができる．

しかし，役目を終えたマルウェアはつねにファイルとして残存しているわけではない．不要になったマルウェアは，攻撃者やマルウェア自身によって削除されることがあるためである．

マルウェアに限らず，1 度削除されたファイルは OS 上からの通常のファイル操作では復元することはできないが，デジタルフォレンジック技術を用いることで復元できる場合がある [1]．ストレージがハードディスクの場合，FAT や NTFS などの Windows 環境においてよく使用されるファイルシステムでは，OS 上からファイルを削除してもファイルのデータ自体はストレージ上に残存する．削除されたファイルは，格納されていた領域の削除フラグが立ち，OS 上から参照できなくなるが，ただちにストレージ上からデータが完全に消去されるわけではない．よって，この領域に格納されているデータを集めて結合することで，元のファイルを復元することができる．

しかし，削除フラグが立った領域は未使用領域として扱われるため，OS 上での新規ファイル生成や既存ファイルへの追記により，この領域に別ファイルのデータが書き込まれることがある．通常，コンピュータの使用にはファイル生成がともなうため，時間の経過とともに削除されたファイルが残存する領域は新しいデータにより上書きされてしまう可能性は高まる [2]．つまり，削除されてからある程度時間が経過してしまったマルウェアは，ストレージから復元しても完全な形で復元することはできず，データの欠損を起こしていることがある．

このようなマルウェアのデータ欠損は，アンチウイルスによるマルウェア同定に著しい悪影響を与えることが著者らの研究で判明している [3]．しかし，攻撃による被害状況を特定するうえでは，データ欠損を起こしたマルウェア（以下，欠損マルウェアと呼ぶ）に対してもマルウェア同定を行えることが望ましい．

そこで本研究では，機械学習を用いてこのような欠損マルウェアの同定を試み，フォレンジックによって復元したマルウェアが欠損していたとしても，元のマルウェア名の同定を可能にする手法を提案する．対象とするマルウェアは x86 向けの Windows 用 PE 形式の 32 bit バイナリプログラムとする．同定対象となる欠損マルウェアは，欠損のないマルウェアを現実環境でのデータ欠損を模倣して人工的に欠損させることで生成する．機械学習を用いてマルウェアの分類を行う際に用いる特徴量は，表層解析情報や挙動解析情報由来の値がよく使われるが，本研究ではマルウェアバイナリをグレースケール画像化したものを使用する．マルウェアおよび欠損マルウェアを画像化したうえでデータセットを生成し，Convolutional Neural Network（以下，CNN と呼ぶ），Triplet Network [4]（以下，TN と呼ぶ），k 近傍法（以下，kNN と呼ぶ）を訓練し，欠損マルウェアから元のマルウェアを同定するモデルを生成する．そして生成したモデルを使って欠損マルウェアの同定を行い，同定精度を示す（実験 1）．また，より現実環境に近づけるためにマルウェアデータセットに良性ソフトウェアを加え，そのうえで改めて良性ソフトウェア・マルウェア判別とマルウェア名同定を行う（実験 2）．

実験 1 の結果，マルウェア訓練データに含まれている無欠損マルウェアを 4,096 bytes 欠損させ同定を試みた場合は約 97.3%，訓練データに含まれていない無欠損マルウェアを同量欠損させ同定を試みた場合は約 48%の同定精度であった．

実験 2 の結果，良性ソフトウェア・マルウェア混合の訓練データに含まれている無欠損検体を 4,096 bytes 欠損させた場合は全マルウェア中の約 85.7%，訓練データに含まれていない無欠損検体を同量欠損させた場合は全マルウェア中の約 38.5%のマルウェアに対して，良性ソフトウェア・マルウェア判別に成功しマルウェア名同定にも成功した．

本論文の貢献は以下のとおりである．

(1) 検体のバイナリデータを画像化したうえで行う機械学習によるマルウェア同定手法は，データ欠損にある程度の耐性があることを示した．
(2) 良性ソフトウェアとマルウェアが混在した環境においても，欠損検体の良性ソフトウェア・マルウェア判別およびマルウェア名同定が可能であることを示した．
(3) 機械学習によるマルウェア名の推定根拠を可視化し，バイナリ中の様々な領域やファイルサイズを推定の手がかりとしていることが欠損耐性の向上に寄与している可能性を示した．

2. 関連研究

欠損のないマルウェアを画像化したうえで分類を試みる関連研究 [5], [6], [7] について述べる．Nataraj らの研究 [5] では，パックされていないマルウェアバイナリをグレースケール画像化し，画像から GIST 特徴量を抽出し k-近傍法を用いてマルウェア分類を行っている．検体数 9,458 個のマルウェア群の分類を試みており，25 ファミリの分類問題で 98%の精度を示した．パックされたマルウェアに対しても同様の実験を行っているが，パックされたマルウェアを元のファミリとは独立した新しいファミリとして扱っている．Hsiao らの研究 [6] では，グレースケール画像化したマルウェアに対して Average Hash [8] を使って画像を

再ラベリングした後，Siamese Network を用いて one-shot 学習を行った．結果，2 クラスの one-shot 問題においては 92%，15 クラスの one-shot 問題では 42%の精度であった．Yakura らの研究 [7] では，パックされたマルウェアを含むマルウェアバイナリ群を画像化し，注意機構を持つ CNN を適用することで，マルウェアの静的解析を効率化する手法を提案している．マルウェア画像を CNN に入力して分類を行う際に高い重要度を持つ画像領域を特定し，該当領域のコードあるいはデータを抽出し提示することで，マルウェア解析作業の効率化を図っている．CNN によるマルウェア分類精度の評価も行っており，542 ファミリの分類問題で Top-1 Error 率（最もあてはまる確率が高いと推定されたクラスが正解クラスと一致していない割合）は 50.97%であった．

マルウェアの欠損が同定に与える影響の研究として，著者らの過去の研究 [3] をあげる．この研究では，マルウェアを自然な形で人工的に欠損させ，欠損がアンチウイルスによるマルウェア同定にどのような影響を与えるかを調査した．その結果，アンチウイルスや欠損箇所にもよるが，欠損前は同定できていた検体に対する同定成功率が，4,096 bytes の欠損により 10%から 80%程度まで低下することを確認した．特にファイル先頭から 4,096 bytes の欠損はアンチウイルスによる同定に致命的な影響を与え，アンチウイルスによらず同定成功率はほぼ 0%まで低下した．

3. 提案手法

3.1 概要

本研究では，欠損を起こした検体のバイナリデータをグレースケール画像化し，その画像を機械学習を用いて分類することを試みる．

通常，機械学習を用いたマルウェア分類においては，マルウェアの表層解析結果や動的解析結果が特徴量としてよく用いられる．検体を表層解析することにより得られるヘッダ情報やインポート情報，検体を挙動解析することで得られる API コール列などは，マルウェアの同定を行ううえでは本来非常に重要な情報である．しかし，検体に欠損が生じている場合，欠損箇所によってはこれらの解析は十分に実行できず，特徴量を取得できない可能性がある．特にファイル先頭付近の欠損は，PE ヘッダなどから得られる有用な表層解析情報を破壊する [3]．一方で，検体のグレースケール画像を特徴量として扱う方式は，検体の PE ファイルとしての整合性が破綻していても特徴量を得ることができるため，欠損に強い方式と考えられる．

単一の検体に対して欠損箇所の数や位置を変えることで複数の欠損検体を生成し，それらを画像化したものを入力データとして TN や CNN を用いた機械学習モデルの訓練および検証を行う．マルウェア名の同定が可能となるよう多クラス分類器としての訓練を行うほか，良性ソフトウェアとマルウェアの判別が可能になるよう二値分類器としての訓練も行う．訓練済みの学習器に対して，新たに生成した欠損検体群を画像化して入力し，その検体がマルウェアであるかの判別やマルウェア名の同定を行う．

3.2 使用する機械学習方式

本節では，提案手法中で使用する機械学習方式について述べる．

3.2.1 Triplet Network

TN は三つ組のサンプルを入力組とするニューラルネットワークであり，サンプル間の関係性を学習することに用いられる．

三つ組の入力は anchor，positive，negative サンプルで構成される．anchor は訓練データの任意のサンプルである．positive・negative サンプルはそれぞれ，anchor サンプルと同・異クラスに属するサンプルである．

訓練時には，この 3 入力をそれぞれベースネットワークに入力して得られる出力に対し，anchor サンプル–positive サンプル間の距離を近く，anchor サンプル–negative サンプル間の距離を小さくするようネットワークを学習する．これにより，サンプル間の関係性を適切に表現する特徴量を出力するニューラルネットワークが構成される．

3.2.2 Convolutional Neural Network

CNN とは，画像解析分野でよく使われているニューラルネットワークである．二次元 CNN の場合は画像の X 軸・Y 軸方向にフィルタをスライドして畳み込み結果を得る．

4. 実験

本章の実験ではまず，マルウェアのみから構成されるデータセットを使用し，人工的に欠損マルウェアを生成し，マルウェア名を同定する機械学習器の訓練および検証を行う（実験 1）．この実験により，欠損マルウェア画像からマルウェア名の同定が可能であるかを調査する．

次に，良性ソフトウェアとマルウェアが混在しているデータセットを使用し，同様に欠損検体を生成し，良性ソフトウェア・マルウェア判別を行う機械学習器の訓練および検証を行う．マルウェアと判別された検体に対して実験 1 と同様のマルウェア同定も実施する（実験 2）．この実験により，より実環境に近い状況で検体が欠損を起こしていても，欠損検体がマルウェアであることの判別とマルウェア名の同定が可能であるかを確かめる．

4.1 データセット生成

本節では，実験において使用する無欠損マルウェア群の詳細，無欠損良性ソフトウェア群の詳細，欠損検体群の生成方法，検体群の画像化方法について説明する．

4.1.1 無欠損マルウェア群

実験に使用する無欠損マルウェア群は，我々が独自に収

表 1 マルウェア検体群情報
Table 1 Dataset statistics of malware.

項目	値
検体数	855 検体
マルウェア名種類数	171 種
パック済検体数	0 検体
サイズの最小値	5,598 bytes
サイズの平均値	約 679,089 bytes
サイズの最大値	4,184,486 bytes
サイズの標準偏差	約 812,625 bytes

集した x86 向けの PE 形式のマルウェア 855 検体である．

検体の情報を**表 1** に示す．各検体の MD5 ハッシュ値はすべて異なる．これら検体群を VirusTotal [9] に与え Microsoft 社のアンチウイルス製品によって，無欠損マルウェア群のマルウェア名 [10] を命名した．本研究では，マルウェア名とはファミリ名だけではなく，Microsoft 社の命名規則において Type から Suffix までのすべてとする．つまり，Type から Suffix まですべて一致するマルウェア群が，同じマルウェア名を持つと定義される．検体群中にマルウェア名は 171 種存在しており，MD5 ハッシュ値は異なるがマルウェア名が同一である検体がそれぞれ 5 検体ずつ含まれている．ファイルサイズの平均値は約 663 KiB で，標準偏差は約 794 KiB である．すべての検体に対して Cylance 社のパッカー検出ツール PyPackerDetect [11] を適用し，パックの痕跡が見つからないことを確認している．ただしこれは，すべての検体が確実にパックされていないことを保証するものではなく，ツールによる検出を免れたパック済み検体が混入している可能性は残っている．

4.1.2 無欠損良性ソフトウェア群

無欠損の良性ソフトウェア検体のデータセットは，日本国内のソフトウェア販売・ダウンロードサイトである Vector [*1] からフリーソフトの収集を行うことで作成した．このとき，ファイルサイズ分布の違いからマルウェアとの区別が容易になることを避けるために，良性ソフトウェア群のファイルサイズ分布がマルウェア群のファイルサイズ分布と近くなるよう選別を行った．

本実験で使用する良性データセットの構成と異なり，実際のストレージ上には実行ファイル以外の様々な形式のファイルが存在している．それを反映し，実行ファイルに限らず文章や画像など様々な形式のファイルを良性データセットに混入させる手法も考えられる．しかし，マルウェアと異なる形式のファイルを混入させた場合，ヘッダやフッタなどの非実行ファイル形式それぞれを特徴付ける箇所が欠損せずに残存すると，機械学習モデルがそれを手がかりに分類を行い，精度が不当に上昇する恐れがある．よって，本研究ではマルウェアと同形式の実行ファイルのみから良性データセットを構成した．

[*1] https://www.vector.co.jp/

表 2 良性ソフトウェア検体群情報
Table 2 Dataset statistics of benign software.

項目	値
検体数	8,676 検体
サイズの最小値	5,632 bytes
サイズの平均値	約 666,960 bytes
サイズの最大値	4,167,350 bytes
サイズの標準偏差	約 926,930 bytes

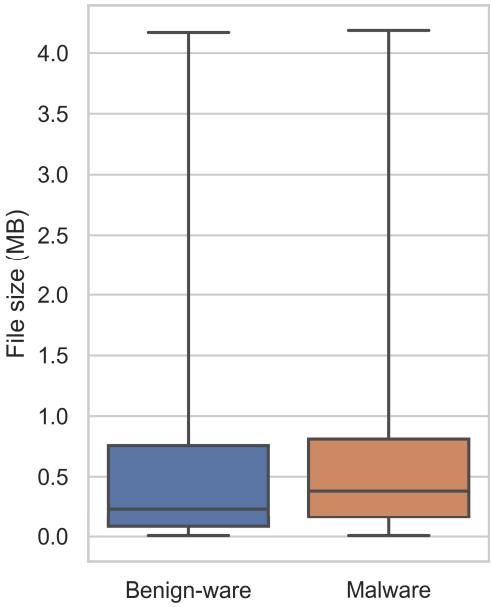

図 1 検体のサイズ比較
Fig. 1 File sizes of samples.

表 2 に実験に使用する良性ソフトウェアデータセットの情報を示す．また，**図 1** に良性ソフトウェア群およびマルウェア群のファイルサイズ分布を箱ひげ図を用いて示す．四角形から伸びている直線の最上部がファイルサイズの最大値，最下部が最小値を表す．四角形の下辺が 25 パーセンタイルを，上辺が 75 パーセンタイルを表す．四角形の内部にある直線は，50 パーセンタイルを表す．良性ソフトウェア，マルウェアともに似たグラフ形状をしており，これは両者のファイルサイズ分布が似通っていることを表している．

実験に使用するすべての良性ソフトウェアは，アンチウイルスソフトによってマルウェアと判定されないことを確認している．

4.1.3 欠損検体群の生成

本項では，無欠損検体群から欠損検体群を生成する方法について述べる．欠損検体群を直接入手しそれに正解ラベルを付与することは容易ではない．そのため，正解ラベルをすでに付与した無欠損検体群を材料とし，自然なデータ欠損を模倣して人工的に欠損検体群を生成することで，その問題を解決する．また，実験 1 の機械学習モデルの訓練および実験 1，2 の検証には，前項までに述べた無欠損検

体群だけではなく，無欠損検体群から生成した欠損検体群も用いる．

著者らが過去に行った調査 [3] によると，NTFS において自然なデータ欠損はクラスタサイズの整数倍アドレスから発生する．当該アドレスを起点に，新しく生成されたファイルデータによる古いファイルデータの上書きが始まる．新しく生成されたファイルデータの書き込み終了後は，書き込み終了アドレスからクラスタサイズの整数倍に到達するまでゼロが書き込まれる．一部の環境ではクラスタサイズ単位ではなくセクタサイズ単位での欠損が起こったが，ゼロ埋めの挙動については変化はなかった．

この結果を受けて，本研究ではクラスタサイズの整数倍のファイルオフセットから，クラスタサイズ分のデータをゼロで上書きすることにより，人工的に欠損検体を生成するものとする．我々の環境ではクラスタサイズは 4,096 bytes であるため，4,096 の整数倍ファイルオフセットから 4,096 bytes 分をゼロで上書きすることを，1 クラスタの欠損と呼称する．欠損クラスタ数 × 4,096 bytes が検体のファイルサイズ以上である場合，その検体を構成する全クラスタが欠損するため，全データがゼロとなる点に留意されたい．

自然なデータ欠損では，欠損した領域がすべてゼロで上書きされるわけではなく，欠損領域の先頭には上書きに使われたファイルのデータが書き込まれている．しかし，これを模して何らかの実ファイルデータで上書きすることで欠損処理を行った場合，その上書きしたデータの内容が実験結果に影響する可能性がある．よって，実ファイルデータによる上書きは採用せず，全検体を一律にゼロで欠損させるものとする．

4.1.4 検体群の画像化

前項までの処理で得た検体群は，画像化処理を施してから機械学習モデルの訓練や検証に使用する．画像化手法は，既存研究 [5], [6] で用いられている手法をおおむね踏襲し，以下のようにする．検体バイナリの各バイトの値を，グレースケール画像における各ピクセルの明度として使用して画像化を行う．値が 0 に近づくほど対応するピクセルは黒くなり，255 に近づくほど白くなる．画像の横幅は，検体バイナリのサイズによって変動させる．10 KiB 未満ならば 32 ピクセル，30 KiB 未満なら 64 ピクセル，60 KiB 未満なら 128 ピクセル，100 KiB 未満なら 256 ピクセル，200 KiB 未満なら 384 ピクセル，500 KiB 未満なら 512 ピクセル，1,000 KiB 未満なら 768 ピクセル，1,000 KiB 以上なら 1,024 ピクセルとした．検体バイナリのサイズが画像の横幅の整数倍と一致しない場合，画像の横幅の整数倍と一致するまでゼロでパディングを行い，グレースケール画像が長方形となるようにした．

このように生成した検体画像を，さらに横幅 200 ピクセル，縦幅 200 ピクセルの固定サイズのマルウェア画像に変換する．元画像サイズが横幅と縦幅ともに 200 ピクセル以下である場合は，全ピクセルの明度が 0 である 200 × 200 ピクセルの背景画像を用意し，背景画像に対して元画像の左上を重ね合わせるように貼り付ける．元画像サイズの横幅または縦幅が 200 ピクセルを超える場合は，縦横比を保持したまま長辺が 200 ピクセルとなるように元画像を縮小する．縮小時の補間手法には BICUBIC 法を使用した．縮小後は，こちらも同様に全ピクセルの明度が 0 である 200 × 200 ピクセルの背景画像を用意し，背景画像に対して元画像の左上を重ね合わせるように貼り付ける．

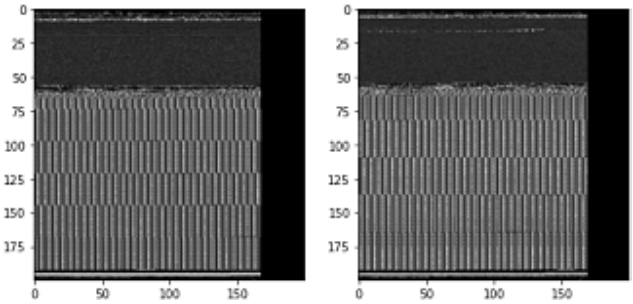

図 2 画像化した Worm:Win32/Gamarue
Fig. 2 Image of Worm:Win32/Gamarue.

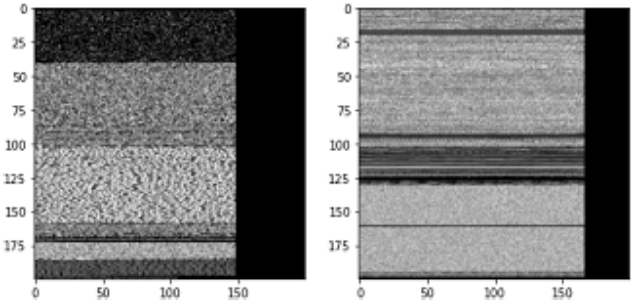

図 3 画像化した Trojan:Win32/Skeeyah.HK!MTB
Fig. 3 Image of Trojan:Win32/Skeeyah.HK!MTB.

この手法で無欠損のマルウェア検体を画像化した例を**図 2** および**図 3** に示す．図 2 は MD5 ハッシュ値の異なる 2 検体のマルウェア（マルウェア名「Worm:Win32/Gamarue」）を画像化した図であり，図 3 はハッシュ値の異なる 2 検体のマルウェア（マルウェア名「Trojan:Win32/Skeeyah.HK!MTB」）を画像化した図である．同一のマルウェア名を持つマルウェア群であっても，図 2 のペアのように似たに画像になる検体もあれば，図 3 のペアのように人間の目には別種のマルウェアのように見える画像になる検体もある．

この問題に対処するために Hsiao ら [6] は画像化した後に，Average Hash [8] を使って同一ファミリ内の画像データセットを似た見た目の画像が集まったグループにさらに分類しているが，我々の研究ではこの操作は行わない．Average Hash による再分類はおそらく実験環境での精度の向上につながるが，実用時も同じマルウェア名かつ同じ見た目のマルウェアを訓練データとして用意しなければな

らなくなるおそれがあり，実用をより困難にするためである．Average Hash による再分類を行わずに高い精度を出せるならば，実用時は画像化時の見た目を意識しなくても同じマルウェア名のマルウェアを訓練データとして用意すれば十分となる．

4.2 実験設計（実験 1・2 共通）

実験 1・2 において使用する機械学習モデルの訓練時は，無欠損検体に対して最低 1 クラスタ，最大 10 クラスタ分の欠損処理を施すことで生成した欠損検体群を使用する．このとき，欠損クラスタ数およびその開始位置は毎回重複を許容しランダムに決定し，動的に欠損検体を生成して訓練に用いる．

両実験においては，4.1.4 項で述べた手法により画像化した検体群を使い，以下の 2 つのシナリオに沿って訓練を行う．

シナリオ 1：訓練および検証時に同一の無欠損検体群を使用して欠損検体をランダム生成し，機械学習モデルへの入力とする．本シナリオで欠損検体から元のマルウェア名を精度良く同定できるということは，ある無欠損マルウェアを訓練データとして保持している場合，そのマルウェアがデータ欠損を起こしても元のマルウェア名を高精度に同定できるということを表す．欠損検体は元となる無欠損検体群から訓練および検証のたびにランダムに生成されるため，シナリオ 1 であっても訓練時と検証時で同一の欠損検体群が使用される可能性は低い．実運用上においてシナリオ 1 が活用できる場面としては，公開情報などを元にして訓練用検体を十分に収集できている場合や，自組織のセキュリティアプライアンスなどの情報から無欠損検体を入手可能であり，その検体が動作していた端末を特定したい場合などがあげられる．

シナリオ 2：各マルウェア名に所属する 5 つの無欠損マルウェアを，訓練時と検証時でそれぞれ 3 : 2 の割合で分割し，それらから欠損マルウェアをランダム生成し，機械学習モデルへの入力とする．良性ソフトウェアを用いる実験 2 では，良性ソフトウェア群も同様の割合で分割を行い訓練と検証に用いる．本シナリオで欠損検体から元のマルウェア名を精度良く同定できるということは，あるマルウェア名に属する無欠損マルウェアを訓練データとして保持している場合，そのマルウェア名を持つ何らかのマルウェアがデータ欠損を起こしても元のマルウェア名を高精度に同定できるということを表す．ハッシュ値が同一の検体を入手することよりもマルウェア名が同一の検体を入手することは容易であるため，実運用上においてシナリオ 2 はシナリオ 1 よりも広い場面で活用が可能である．

4.3 マルウェア同定機械学習モデル

4.2 節で定義したシナリオに従い，以下のように TN および CNN，比較用としてニューラルネットワークを用いない機械学習方式である k 近傍法の訓練を行う．これらの機械学習モデルを，マルウェア名の同定に使用する．

4.3.1 Triplet Network の構成

図 4 に実験 1 で使用する TN の構造を示す．anchor サンプル，positive サンプル，negative サンプルがそれぞれベースネットワークを通りベクトル化され，Triplet Loss 損失関数により損失が計算される．3 つのベースネットワークは重みを共有している．

表 3 にベースネットワークの構造を示す．この構造の前半部分は文献 [6] で使われているものと同一である．各二次元畳み込み層（Conv2D）の活性化関数には Rectified Linear Unit（relu）を使用した．末尾の Lambda では L2 正規化を行っている．オプティマイザには Adam（学習率 0.001）を利用した．

anchor サンプルとして欠損マルウェア画像を，positive サンプルとして anchor サンプルと同一のマルウェア名を持つ無欠損マルウェア画像を，negative サンプルとして anchor サンプルと異なるマルウェア名を持つ無欠損マルウェア画像を与える．より詳細には，訓練データセットと

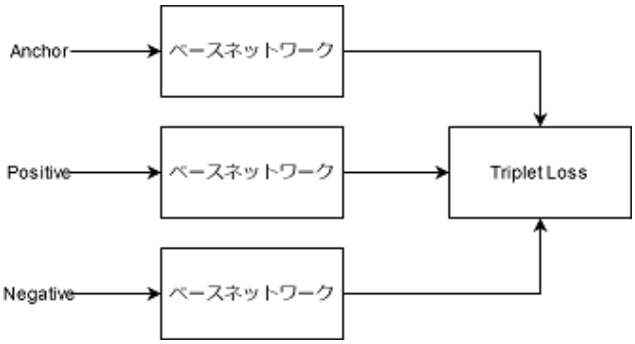

図 4 Triplet Network の構造
Fig. 4 Structure of the triplet network.

表 3 ベースネットワークの構造
各サイズ中の数字は，順にバッチサイズ（つねに任意長），マップ高さ，マップ幅，チャネル数を表す．
Table 3 Structure of the base network.

レイヤ	入力サイズ	出力サイズ
Input	(n, 200, 200, 1)	(n, 200, 200, 1)
Conv2D	(n, 200, 200, 1)	(n, 191, 191, 64)
MaxPooling2D	(n, 191, 191, 64)	(n, 95, 95, 64)
Conv2D	(n, 95, 95, 64)	(n, 89, 89, 128)
MaxPooling2D	(n, 89, 89, 128)	(n, 44, 44, 128)
Conv2D	(n, 44, 44, 128)	(n, 41, 41, 128)
MaxPooling2D	(n, 41, 41, 128)	(n, 20, 20, 128)
Conv2D	(n, 20, 20, 128)	(n, 17, 17, 256)
MaxPooling2D	(n, 17, 17, 256)	(n, 8, 8, 256)
Flatten	(n, 8, 8, 256)	(n, 16,384)
Dense	(n, 16,384)	(n, 100)
Lambda	(n, 100)	(n, 100)

して，各マルウェア名ごとに 30 パターンの anchor サンプル–positive サンプルのペアを，1 パターンの anchor サンプル–posirive サンプルのペアごとに 5 つの negative サンプルを毎回ランダム生成した．このように，エポックごとに毎回 anchor サンプル–positive サンプル–negative サンプルの三つ組をランダム生成し，Triplet Network の訓練に使用する．以上の条件で，Triplet Network の訓練を 100 エポック実施した．

TN の訓練終了後，訓練済みのベースネットワークに入力を伝播して得られるベクトルを特徴量とし，クラス分類用のモデルを訓練する．クラス分類用のモデルとして今回は，中間層 1 層（ユニット数 128）の 3 層ニューラルネットワークを利用した．

4.3.2 Convolutional Neural Network の構成

実験 1 および 2 で使用する CNN の構造は，Triplet Network のベースネットワークである表 3 の構造とほぼ同一である．表 3 末尾の Lambda レイヤの後ろに，マルウェア名のクラス数と同数である 171 ユニットの全結合層（Dense）を追加し，活性化関数として softmax 関数を使用した．オプティマイザは TN と同様に Adam（学習率 0.001）とした．

各マルウェア名ごとに 5 つ（シナリオ 1）または 3 つ（シナリオ 2）の無欠損検体画像，1 つの無欠損検体ごとに 5 パターンの欠損マルウェア画像をランダム生成し，訓練データセットとした．すなわち，1 マルウェア名ごとに 30 パターン（シナリオ 1）または 18 パターン（シナリオ 2）のマルウェア画像が生成される．TN での実験と同様に，エポックごとにこれらの訓練データセットはランダムに再生成され，順次 CNN へ入力される．以上の条件で，訓練を 100 エポック実施した．

4.3.3 k 近傍法

ニューラルネットワークを用いない機械学習手法として，実験 1 では k 近傍法（kNN）による分類も実施する．訓練時には訓練データセットに含まれる各グレースケール画像における各ピクセルの明度情報を特徴空間に配置し，検証時には検証対象データと近い位置に配置された k 個の訓練データのラベルを参照して多数決をとることでクラス分類を行う方式である．使用するデータセットは前項の CNN と同様とする．k の値はシナリオ 1 の場合は 5，シナリオ 2 の場合は 3 とした．

4.4 実験 1

本節では，マルウェアのみから構成されるデータセットを使用し，機械学習により欠損マルウェアのマルウェア名同定を行う実験について詳細を述べる．

4.4.1 実験 1 の設計と評価基準

実験 1 で使用するデータセットは 4.1.1 項で生成したマルウェアデータセットである．4.2 節の 2 つのシナリオともに，検証時には検証用の無欠損マルウェア検体群に対して，1 検体ごとに 0 から 9，10，20，30，40，50 カ所のクラスタをランダムに欠損させてから画像化を行うという処理を 10 回繰り返し，検証用の欠損検体画像群を生成する．つまり，1 つのマルウェア検体あたり欠損箇所数ごとに 10 枚の欠損マルウェア画像が得られる．欠損箇所は 1 つの欠損マルウェアの中で重複が起こらないようにランダムに決定する．

また，著者らの過去の研究 [3] で，マルウェアの先頭クラスタがアンチウイルスによる同定に致命的な悪影響を及ぼすことが分かっている．このような先頭クラスタ欠損に対する耐性を確かめるためのデータセットも別途生成した．

検証用の無欠損マルウェア群に対してランダムなクラスタ欠損処理を施したデータセットを，検証データセット A と呼称する．また，検証用の無欠損マルウェア群の先頭クラスタを必ず欠損させ，かつ前述のランダムなクラスタ欠損処理を施したデータセットを，検証データセット B と呼称する．

このようにして生成した検証データセット A および B を訓練したモデルに入力し，マルウェア名 171 クラスの分類問題をどれだけの精度で解くことができるか検証する．

評価基準として，同定精度を用いる．これは，検証用のすべてのマルウェアのうち，マルウェアを欠損させて機械学習モデルに入力したとき，元のマルウェア名を 171 種のマルウェア名の中から当てることのできた検体の割合である．

4.4.2 実験 1 の結果

図 5 および**表 4** 前半に，検証データセット A に対して TN と CNN，kNN を用いた，欠損マルウェア同定の結果

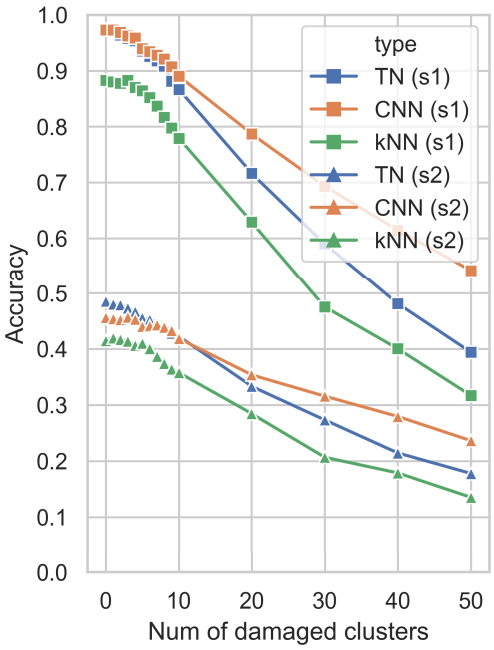

図 5 全箇所をランダムに欠損させた場合の同定精度
Fig. 5 Accuracy of malware name identification in the case that clusters are randomly deleted.

表 4 同定精度の詳細
Table 4 Detailed accuracy of malware name identification.

全箇所をランダムに欠損させた場合のマルウェア名同定精度（検証データセット A を使用）															
	欠損クラスタ数														
	0	1	2	3	4	5	6	7	8	9	10	20	30	40	50
TN (s1)	0.974	0.973	0.964	0.958	0.954	0.935	0.926	0.918	0.908	0.881	0.867	0.716	0.590	0.481	0.394
CNN (s1)	0.973	0.973	0.969	0.962	0.958	0.939	0.934	0.928	0.921	0.907	0.890	0.787	0.693	0.614	0.542
kNN (s1)	0.883	0.880	0.877	0.883	0.870	0.864	0.852	0.837	0.816	0.798	0.779	0.629	0.476	0.400	0.317
TN (s2)	0.485	0.480	0.479	0.473	0.467	0.456	0.452	0.449	0.445	0.428	0.423	0.333	0.273	0.214	0.178
CNN (s2)	0.456	0.454	0.453	0.457	0.453	0.440	0.442	0.443	0.439	0.433	0.419	0.354	0.316	0.280	0.236
kNN (s2)	0.415	0.420	0.417	0.414	0.407	0.410	0.401	0.387	0.374	0.364	0.358	0.285	0.207	0.179	0.135
先頭クラスタを必ず欠損させた場合のマルウェア名同定精度（検証データセット B を使用）															
	先頭クラスタ以外の欠損クラスタ数														
	0	1	2	3	4	5	6	7	8	9	10	20	30	40	50
TN (s1)	0.972	0.969	0.963	0.959	0.951	0.933	0.927	0.916	0.903	0.885	0.869	0.724	0.587	0.478	0.396
CNN (s1)	0.972	0.970	0.967	0.961	0.953	0.939	0.932	0.927	0.920	0.905	0.890	0.786	0.695	0.616	0.540
kNN (s1)	0.882	0.887	0.886	0.885	0.880	0.866	0.857	0.839	0.822	0.798	0.784	0.622	0.479	0.400	0.316
TN (s2)	0.491	0.480	0.476	0.470	0.462	0.455	0.453	0.449	0.444	0.435	0.415	0.334	0.269	0.213	0.177
CNN (s2)	0.453	0.453	0.453	0.453	0.453	0.442	0.439	0.439	0.439	0.430	0.419	0.354	0.317	0.273	0.236
kNN (s2)	0.421	0.422	0.419	0.418	0.410	0.405	0.396	0.393	0.381	0.369	0.362	0.287	0.206	0.176	0.137

を示す．図表中において，カッコ内が s1 である場合はシナリオ 1，すなわち訓練時と検証で同じ無欠損マルウェア群から欠損マルウェア群をランダム生成したことを表す．カッコ内が s2 である場合はシナリオ 2，すなわち訓練時と検証でそれぞれ別の無欠損マルウェア群から欠損マルウェア群をランダム生成したことを表す．

シナリオ 1 の場合，欠損クラスタ数が 8 クラスタ以下であるならば，TN，CNN ともに 90%以上の精度でマルウェア名の同定ができている．シナリオ 2 の場合，欠損クラスタ数が 10 クラスタ以下であるならば TN，CNN ともに 41%から 48%程度の精度でマルウェア名の同定ができている．

欠損箇所が少ないうちは TN と CNN で同定精度にあまり差はないが，欠損箇所が増えるにつれ CNN の同定精度が勝る．また，どちらのシナリオにおいても kNN は同定精度において TN と CNN に劣る結果となった．

図 6 および表 4 後半に，検証データセット B に対して TN と CNN，kNN を用いた，欠損マルウェア同定の結果を示す．

先頭クラスタの欠損はアンチウイルスによる同定には致命的な悪影響を与え同定成功率をほぼ 0%まで押し下げたが，機械学習を用いた本手法においてはさほど重要ではなく，検証データセット A とほぼ同様の同定精度となった．

シナリオ 1 では，先頭クラスタが欠損していたとしても，ニューラルネットワークを使った機械学習による同定により 97.2%の精度でマルウェア名の同定に成功している．シナリオ 2 でも，TN により 49.1%の精度でマルウェア名の同定に成功している．

図 7 に，訓練後の TN のベースネットワークに無欠損マルウェア画像 855 枚 171 クラスを入力して得られる 100 次元の特徴量を，t-SNE [12] を用いて 2 次元まで次元削減し可視化した図を示す．1 次元目を横軸，2 次元目を縦軸に配置する．マルウェア名が同一の検体に対して同じ色を割り当てて図示している．図中において距離が近いほど，特徴量の傾向が近いことを表している．

同じマルウェア名を持つ検体の特徴量間の距離は近く，異なるマルウェア名を持つ検体の特徴量間の距離は遠くなっており，訓練済みの TN により得られる特徴量は，それを入力として受け取る分類器の精度向上に寄与している

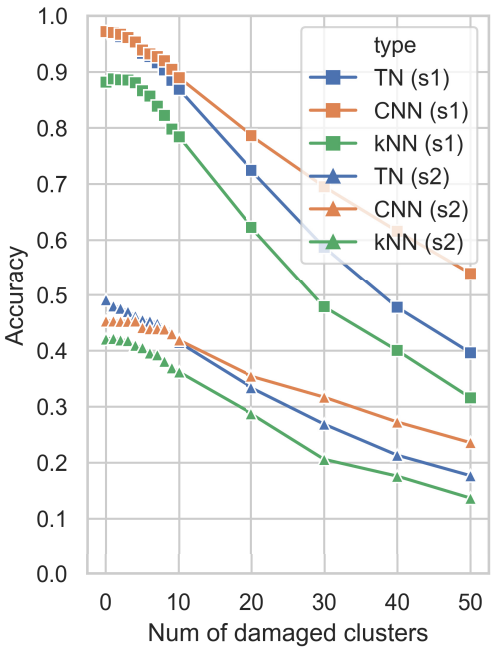

図 6 先頭クラスタを必ず欠損させた場合の同定精度
Fig. 6 Accuracy of malware name identification in the case that the first cluster is bound to be deleted.

表 5 良性ソフトウェア・マルウェア判別モデルの各評価指標（詳細）
Table 5 Evaluation metrics of binary classification between benign software and malware.

		欠損クラスタ数														
		0	1	2	3	4	5	6	7	8	9	10	20	30	40	50
Acc.	s1	0.917	0.920	0.926	0.926	0.923	0.914	0.910	0.901	0.892	0.885	0.873	0.788	0.717	0.684	0.657
Acc.	s2	0.760	0.760	0.761	0.760	0.753	0.750	0.745	0.744	0.738	0.729	0.720	0.665	0.619	0.612	0.590
Pre.	s1	0.956	0.951	0.949	0.945	0.940	0.936	0.932	0.923	0.920	0.917	0.912	0.863	0.816	0.828	0.825
Pre.	s2	0.885	0.878	0.866	0.858	0.852	0.856	0.850	0.851	0.848	0.836	0.824	0.779	0.739	0.755	0.763
Rec.	s1	0.873	0.884	0.900	0.904	0.902	0.887	0.882	0.874	0.858	0.844	0.824	0.682	0.555	0.459	0.392
Rec.	s2	0.594	0.601	0.613	0.618	0.607	0.597	0.589	0.587	0.577	0.565	0.554	0.453	0.359	0.322	0.251

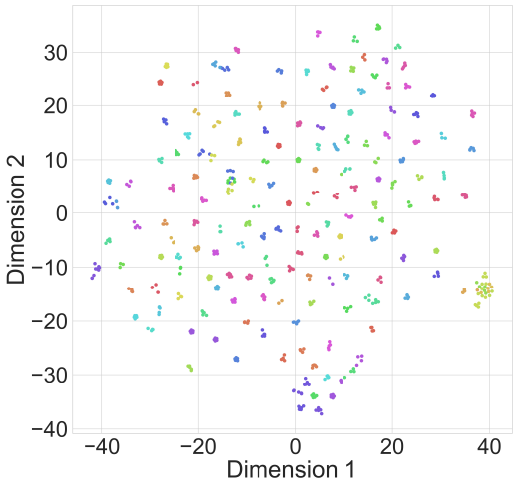

図 7 Triplet Network により得られた特徴量の分布
Fig. 7 Distribution of features obtained by triplet network.

ことが分かる．

4.5 実験 2

4.4 節では，欠損ソフトウェアが何らかのマルウェアであると断定可能な状況を想定して同定実験を行った．しかし，実際のストレージ上で欠損を起こしているソフトウェアは必ずしもマルウェアとは限らず，悪性動作を行わない良性ソフトウェアである可能性がある．

本章では，データセットに良性ソフトウェアとマルウェアが混在している状況を想定し，このような状況で欠損マルウェアのマルウェア名同定を行う実験について述べる．

4.5.1 実験 2 の設計と評価基準

実験 2 で扱うデータセットは，4.1.1 項で生成したマルウェアデータセットと 4.1.2 項で生成した良性ソフトウェアデータセットを連結し生成したものである．このデータセットにおいては，マルウェアと良性ソフトウェアで著しく検体数に差がある．そのため，欠損処理時にオーバサンプリングを行う．欠損処理時に良性ソフトウェアに対しては無欠損検体 1 種から欠損検体 1 種をランダム生成し，マルウェアに対しては無欠損検体 1 種から欠損検体 10 種をランダム生成することで，それぞれの欠損検体の数をほぼ同数とした．

実験 2 ではまず，このデータセットに属する検体に欠損処理を施したうえで，後述の良性ソフトウェア・マルウェア判別を実施する．そして，この判別によってマルウェアと判定された検体に対して，実験 1 で良い成績を残した CNN を用いてマルウェア名同定を行う．

良性ソフトウェア・マルウェア判別モデル：欠損検体がマルウェアであると断定できない場合，マルウェア名同定を行う前にその検体がマルウェアであるかどうかを判別する必要がある．良性ソフトウェア・マルウェアを判別する二値分類を行うために，実験 1 でのマルウェア名同定において良い成績を残した CNN のネットワーク構造を流用し新たに機械学習器を構成した．具体的には，表 3 末尾の Lambda の後ろにユニット数 1 の全結合層を追加し活性化関数として sigmoid 関数を採用し，二値分類に適した形状とした機械学習器である．このように構成した CNN に対し，欠損処理を施したデータセットを入力することで訓練および検証を行う．良性ソフトウェア・マルウェア判別モデルの訓練時においては，実験 1 のマルウェア名同定モデルの訓練時とは異なり，無欠損検体画像は訓練データセットに含んでいない．

良性ソフトウェア・マルウェア判別モデルの評価基準として，Recall，Accuracy，Precision を用いる．ここで，Accuracy（Acc.）は全検体のうち，良性ソフトウェア・マルウェア判別モデルによって良性ソフトウェアかマルウェアかを正答できた検体の割合である．Precision（Pre.）は良性ソフトウェア・マルウェア判別モデルによってマルウェアと判定された検体のうち，実際にマルウェアである検体の割合である．Recall（Rec.）は実際にマルウェアである検体のうち，良性ソフトウェア・マルウェア判別モデルによってマルウェアと判定された検体の割合である．

4.5.2 実験 2 の結果

表 5 および**図 8** に良性ソフトウェア・マルウェア判別モデルの評価指標値を示す．

欠損箇所が 10 クラスタ以下の場合は，シナリオ 1 では Precision が約 0.873 以上，シナリオ 2 では 0.720 以上あり，良性ソフトウェア・マルウェア判別においてマルウェアと判定された場合は高い割合でマルウェアである．10 クラス

表 6 マルウェアと判定された検体のうちマルウェア名同定に成功した割合（詳細）
Table 6 Accuracy of malware name identification on the samples predicted as malware.

	欠損クラスタ数														
	0	1	2	3	4	5	6	7	8	9	10	20	30	40	50
s1	0.927	0.922	0.921	0.915	0.912	0.907	0.905	0.893	0.889	0.885	0.881	0.805	0.734	0.716	0.675
s2	0.562	0.563	0.554	0.543	0.549	0.540	0.537	0.534	0.533	0.521	0.510	0.441	0.412	0.387	0.393

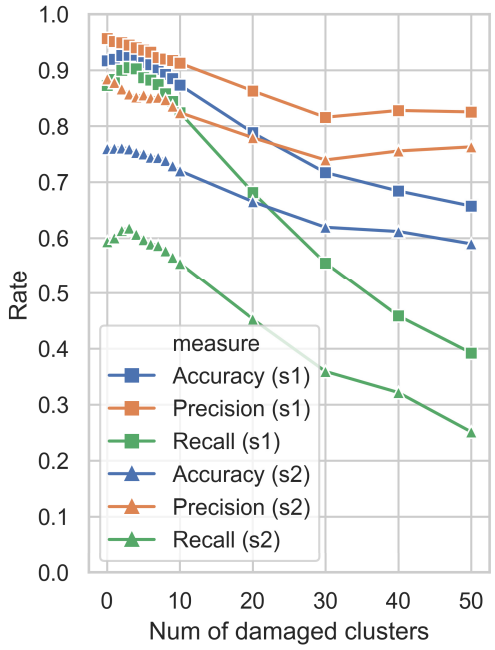

図 8 良性ソフトウェア・マルウェア判別モデルの各評価指標
Fig. 8 Evaluation metrics of binary classification between benign software and malware.

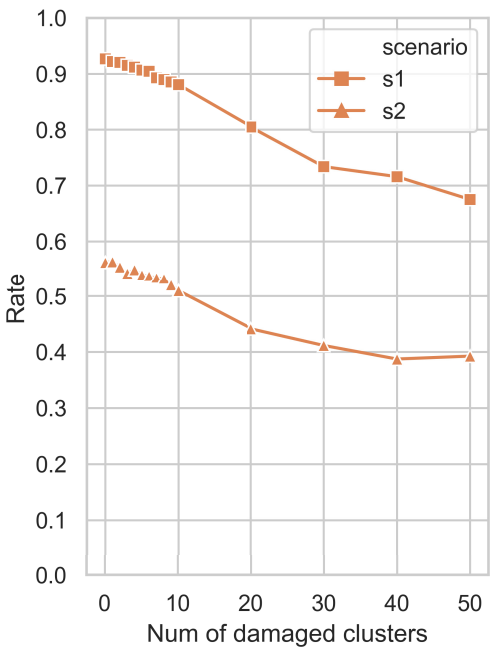

図 9 マルウェアと判定された検体のうちマルウェア名同定に成功した割合
Fig. 9 Accuracy of malware name identification on the samples predicted as malware.

タ超の欠損であっても，Precision はどちらのシナリオでも 0.75 以上を維持しており，マルウェアと判定された検体が実際にマルウェアである割合は依然として高い．

ただし，Recall は欠損箇所の増加に従って減少する傾向であり，欠損による検知逃れの増加がみられる．50 クラスタの欠損で，シナリオ 1 では約 60.8%，シナリオ 2 では約 74.9%のマルウェアを誤って良性ソフトウェアと判定している．

シナリオ 2 はシナリオ 1 に比べ各評価指標値の減少がみられるが，欠損が各評価指標に与える影響については両シナリオで同様の傾向を示している．

無欠損の検体の評価指標が微量の欠損が起きている検体に劣っている理由は，良性ソフトウェア・マルウェア判別モデルにおいて無欠損検体を訓練データとして使用していないためと考えられる．

表 6 および**図 9** に，全検体のうち良性ソフトウェア・マルウェア判別モデルによってマルウェアと判定された検体に対して，実験 1 と同一のマルウェア名同定を行った結果を示す．ここで，実際は良性ソフトウェアであるにもかかわらず良性ソフトウェア・マルウェア判別モデルによってマルウェアと判定された検体は，つねにマルウェア名同定に失敗したものとして扱っている．

欠損が進むにつれ同定精度が下がる点は実験 1 でのマルウェア名同定と同様である．しかし，50 カ所の欠損であっても同定精度はシナリオ 1 で 67%超，シナリオ 2 で 39%超と，実験 1 と比較して高い値となっている．これは，欠損によって分類困難となるマルウェアが，良性ソフトウェア・マルウェア判別モデルで誤って良性ソフトウェアと判定されてしまい，マルウェア名同定の対象となっていないためと考えられる．

表 7 および**図 10** に，全マルウェアのうち良性ソフトウェア・マルウェア判別モデルによってマルウェアと判定されマルウェア名同定にも成功した検体の割合を示す．欠損箇所が 10 クラスタ未満の場合，シナリオ 1 では全マルウェア中 81.5%から 87.6%の検体をマルウェアと判定しかつマルウェア名の同定にも成功している．同条件でシナリオ 2 では，34.3%から 39.2%の検体をマルウェアと判定しかつマルウェア名の同定に成功している．

5. 考察

5.1 データ欠損がマルウェア名の同定に与える影響

実験 1 の結果から，データセット中にマルウェアのみ存

表 7 全マルウェアのうち，良性ソフトウェア・マルウェア判定とマルウェア名同定の双方に成功した検体の割合（詳細）

Table 7 Percentage of malware samples such that both benign-software/malware classification and malware name identification were succeeded.

	欠損クラスタ数														
	0	1	2	3	4	5	6	7	8	9	10	20	30	40	50
s1	0.846	0.857	0.873	0.876	0.876	0.859	0.856	0.846	0.829	0.815	0.795	0.636	0.499	0.397	0.321
s2	0.377	0.385	0.392	0.391	0.391	0.377	0.372	0.368	0.363	0.352	0.343	0.257	0.200	0.165	0.129

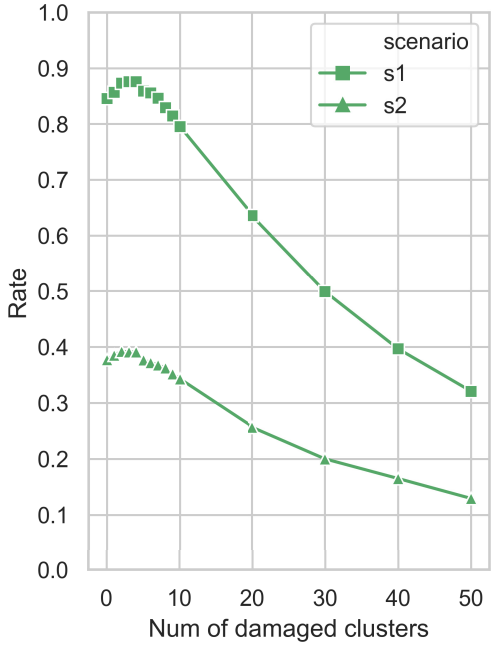

図 10 全マルウェアのうち，良性ソフトウェア・マルウェア判定とマルウェア名同定の双方に成功した検体の割合

Fig. 10 Percentage of malware samples such that both benign-software/malware classification and malware name identification were succeeded.

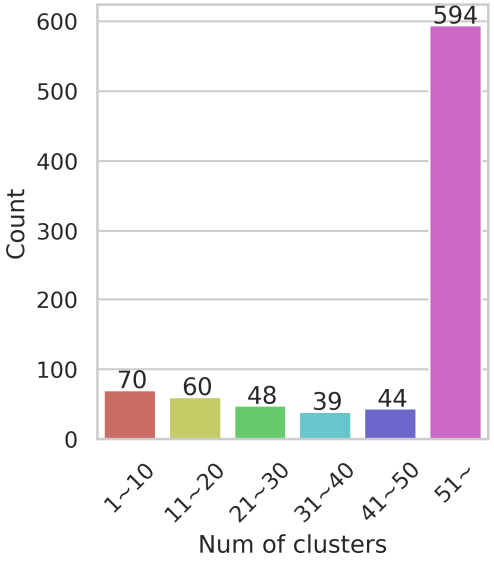

図 11 無欠損検体のクラスタ数

Fig. 11 The number of clusters of complete samples.

在する場合，TN および CNN は欠損マルウェアを高精度に同定できており，アンチウイルスを使用した同定 [3] よりも欠損の影響を低く抑えられていることが分かる．

シナリオ 2 の条件であっても，10 クラスタ程度の欠損までであれば 171 クラス分類問題を約 42%超の精度で解けている．これはあるマルウェア名を持つ無欠損マルウェアを訓練データとして保持していれば，同名かつハッシュが異なるマルウェアが 10 クラスタ欠損した場合でも元のマルウェア名を 42%前後の精度で推定できる可能性を示す．

シナリオ 1 およびシナリオ 2 ともに，データ欠損箇所が増えるにつれて同定精度が低下していることが見て取れる．これは，欠損クラスタ数が増えると同定に利用できる情報が減るため精度が低下していると考えられる．また，欠損箇所の合計サイズが検体のサイズを上回った場合，検体の情報が完全に消えることも影響している．

図 11 に無欠損検体のクラスタ数を示す．サイズが 50 クラスタ未満である検体も多く全体の約 3 割を占め，今回の実験条件ではデータ欠損によりこれら検体の情報は全損しうる．サイズの小さい検体は少量の上書きでデータの全損が起こりやすいため，欠損が起きた場合に同定が難しくなる検体であるといえる．10，20，30，40，50 クラスタの欠損が生じると，全検体はそれぞれ平均して 76.9%，64.2%，55.2%，48.0%，41.8%のデータが残存する．データが全損した検体に対してもマルウェア名同定は行われるが，これは同条件でデータが全損する検体の中からランダムにラベルを推測しているにすぎない．

実験 1 では，多くの実験条件において CNN の精度が TN の精度を上回った．TN は訓練データの各クラスに属するサンプル数が少なくても各クラスを精度良く分離できる手法として知られている．訓練データにおける無欠損マルウェアの各クラスあたりのサンプル数はシナリオ 1 で 5 検体，シナリオ 2 で 3 検体と少数であったが，ランダム箇所欠損により各クラスに属する欠損マルウェアサンプル数は膨大になるため，TN の優位性が活きなかったと考えられる．

5.2 マルウェア名同定の推定根拠の調査

機械学習による同定手法が欠損に対してある程度の耐性を得た原因を探るために，検体のどの部分に着目してクラス推定が行われたかを調査した．Grad-CAM [13] を用いる

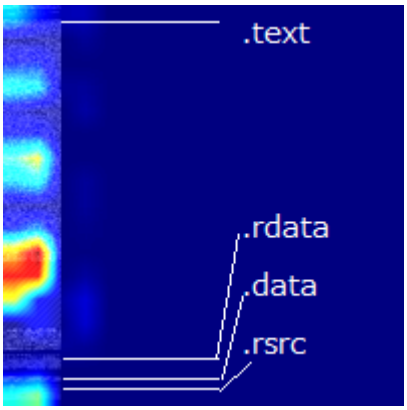

図 12 推定根拠の可視化の一例 （Trojan:Win32/Dorv.A）
モデルが推定時に強く注視した箇所ほど赤く示す

Fig. 12 An example of visual explanation for malware identification (Trojan:Win32/Dorv.A).

ことで，訓練済みモデルに対して画像を入力しクラス推定を行った際，推定に強く寄与した画像の領域をハイライトすることができる．

シナリオ 1 の条件で訓練を行った CNN に対して，Grad-CAM を用いて推定根拠の可視化を行った．結果の一例として，無欠損マルウェア検体 Trojan:Win32/Dorv.A をシナリオ 1 の CNN を用いてクラス推定を行った際の推定根拠を**図 12** に示す．主にファイル先頭の PE ヘッダ領域，.text セクション，.rsrc セクションを推定根拠としている．推定根拠をファイルの一部分に依存しすぎず，ファイル全体の各所を手がかりとしているため，局所的な欠損に耐性があると思われる．また，検体を画像化する際にゼロパディングが行われた部分も，推定根拠として使用されていることが分かる．ゼロパディング領域の形状は検体のサイズに依存するため，実質的に検体サイズを推定根拠として使用していると考えられる．検体サイズは攻撃者やマルウェア自身によって容易に変更可能であるため，この性質は好ましくない．

5.3 データ欠損がソフトウェアの良性悪性判定およびマルウェア名同定に与える影響

実験 2 の結果から，データセット中に欠損良性ソフトウェアと欠損マルウェアが混在している場合においても，CNN を用いた欠損検体の良性ソフトウェア・マルウェア判別とマルウェア名同定は可能であると考えられる．10 クラスタ未満の欠損のとき，シナリオ 1 で 81.5%超の割合のマルウェアを正しくマルウェアと判定し，マルウェア名の同定にも成功している．欠損箇所が増えるにつれ精度低下が起こる点は実験 1 と共通である．10 クラスタ未満の欠損のとき，シナリオ 2 では 35.2%超の割合のマルウェアを正しくマルウェアと判定し，マルウェア名の同定にも成功している．シナリオ 2 の条件で実運用するためには，同一マルウェアに侵害されたと思われるストレージを複数用意しそれぞれに本手法を適用して多数決をとることや，侵害の発覚のきっかけになった挙動を手がかりにマルウェア名の候補を絞るなど，精度向上のための施策の検討が必要である．

良性ソフトウェア・マルウェア判別モデル単体で見た場合，欠損が増えても Presicion はさほど低下しないが，Recall の急な低下がみられる．これは，今回生成した良性ソフトウェア・マルウェア判別モデルは検体の欠損箇所が多いとき，検体を良性ソフトウェアと判定しやすいということを表している．

5.4 本手法が要求するフォレンジックツールの能力

実際の運用の場において本手法を適用するためには，デジタルフォレンジック技術により欠損マルウェアを復元する必要がある．ここで，本手法がフォレンジックツールに対して求める能力について考察を行う．ファイル消去が起こったとき，NTFS においてファイルデータのストレージ上の位置を記録しているメタデータが残存している場合は，フォレンジックツールにはこのメタデータを読み取りファイルを復元する能力が求められる．上述のメタデータが残存していない場合，フォレンジックツールにはストレージ上を走査して，ファイルヘッダなどを手がかりにファイルを復元する能力が求められる．メタデータの残存もなく手がかりとなる領域も欠損を起こしている場合，ストレージ上のどの領域が 1 つのファイルであるかを特定することは難しくなる．前述の条件でもファイルの復元が実施できる能力を持つフォレンジックツールを使用すれば，そのような状況でも本手法の適用は可能である．

5.5 マルウェアのファイル入出力挙動が本手法へ与える影響

マルウェアによっては，元ファイルの情報をストレージ上から完全に消去する専用ソフトウェアを使用して自分自身を削除することがある [14]．そのようなツールを使用された場合はマルウェアのすべての領域が欠損するため，本手法の適用はできない．

ランサムウェアなどの大量のファイル入出力をともなうマルウェアの場合，消去されたファイルの欠損がより促進されてしまうことも考えられる．悪性挙動である大量のファイル入出力を担当するマルウェアモジュールが自身を消去するタイミングは，通常自身の悪性挙動を終えた後であるため，このモジュールに関しては欠損が促進される恐れは少ない．しかし，ダウンローダーやドロッパなど悪性挙動のための準備を行うマルウェアモジュールは，二次検体が行う大量のファイル入出力をともなう悪性挙動が開始される前に自分自身を消去することもある．この場合，悪性挙動によって欠損が促進される可能性が高まり，結果として欠損箇所の増加を招くため本手法による同定精度は低

下すると予想される．

6. まとめと今後の課題

本論文では，ファイルが消去されたことによりデータが欠損したマルウェアに着目し，機械学習を用いてこのような欠損マルウェアから元のマルウェア名を同定する手法について実験を行った．

1 クラスタ（4,096 bytes）程度の欠損であれば 97%前後（シナリオ 1：無欠損時にハッシュ値が同一である検体を訓練データに含めた場合），もしくは 48%前後（シナリオ 2：無欠損時にハッシュ値は異なるが名称は同一である検体を訓練データに含めた場合）の精度で元のマルウェア名を同定できることが分かった．先頭クラスタの欠損はアンチウイルスによるマルウェア名同定の精度をほぼ 0%まで押し下げるが，機械学習による同定手法の場合はほぼ影響はなかった．

また，良性ソフトウェアとマルウェアが混在したより実環境に近い環境において，欠損したバイナリの良性ソフトウェア・マルウェア判別を行いかつマルウェア名の同定も行う手法についても実験を行った．

10 クラスタ未満の欠損であればシナリオ 1 で 81.5%超，シナリオ 2 で 35.2%超の欠損マルウェアにおいて，良性ソフトウェア・マルウェア判定およびマルウェア名の同定に成功した．

今後の課題として，悪性データセットの検体数を拡充し，さらにパック済みマルウェアや x86_64 向けマルウェアを含めることがあげられる．特にパックはバイナリを画像化したときの見た目を著しく変えるため，シナリオ 2 での分類精度を押し下げると予想される．現時点でのシナリオ 2 の分類精度に関してもまだ十分とはいえないため，モデルの改善も課題である．今回使用していない機械学習技術の適用やハイパーパラメータの調整により，分類精度の向上を図る．

さらに，良性データセットに画像ファイルなどの非実行形式の良性ファイル群を加え，その影響を観測することも課題としてあげられる．

また，本研究はデータの欠損箇所をゼロで埋めたが，ランダムデータあるいは別のソフトウェアのデータで埋めた場合，良性ソフトウェア・マルウェア判別やマルウェア名同定の精度にどのような影響が出るかを確かめることも，今後の課題である．

参考文献

[1] Hand, S., Lin, Z., Gu, G. and Thuraisingham, B.: BinCarver: Automatic Recovery of Binary Executable Files, *Proc. 12th Annual Digital Forensics Research Conference* (*DFRWS '12*) (2012).

[2] 林 健，佐々木良一：時間経過に着目した HDD のデータ復元に関する実験と解析，情報処理学会研究報告，Vol.2013-CSEC-60, No.14 (2013).

[3] 小久保博崇，大山恵弘：マルウェア検体のデータ欠損がマルウェア同定に与える影響の調査，コンピュータセキュリティシンポジウム 2019 論文集，No.2019, pp.947–952（オンライン），入手先 ⟨https://ci.nii.ac.jp/naid/170000181104/⟩(2019).

[4] Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B. and Wu, Y.: Learning Fine-Grained Image Similarity with Deep Ranking, *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp.1386–1393 (2014).

[5] Nataraj, L., Karthikeyan, S., Jacob, G. and Manjunath, B.S.: Malware Images: Visualization and Automatic Classification, (online), DOI: 10.1145/2016904.2016908 (2011).

[6] Hsiao, S.-C., Kao, D.-Y., Liu, Z.-Y. and Tso, R.: Malware Image Classification Using One-Shot Learning with Siamese Networks, *Procedia Computer Science*, Vol.159, pp.1863–1871 (online), DOI: 10.1016/j.procs.2019.09.358 (2019).

[7] Yakura, H., Shinozaki, S., Nishimura, R., Oyama, Y. and Sakuma, J.: Neural malware analysis with attention mechanism, *Computers & Security*, Vol.87, p.101592 (2019).

[8] Imagehash: Imagehash, available from ⟨https://github.com/bjlittle/imagehash/⟩.

[9] ChronicleSecurity: VirusTotal, available from ⟨https://www.virustotal.com/⟩.

[10] Microsoft: Malware names, available from ⟨https://docs.microsoft.com/en-us/windows/security/threat-protection/intelligence/malware-naming⟩.

[11] Cylance: PyPackerDetect, available from ⟨https://github.com/cylance/PyPackerDetect⟩.

[12] van der Maaten, L. and Hinton, G.: Visualizing Data using t-SNE, *Journal of Machine Learning Research*, Vol.9, pp.2579–2605 (online), available from ⟨http://www.jmlr.org/papers/v9/vandermaaten08a.html⟩ (2008).

[13] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization, *2017 IEEE International Conference on Computer Vision* (*ICCV*), pp.618–626 (2017).

[14] The MITRE Corporation: Indicator Removal on Host: File Deletion, available from ⟨https://attack.mitre.org/techniques/T1070/004/⟩.

小久保 博崇

2012 年筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻修了．同年より株式会社富士通研究所所属．サイバーセキュリティに関する研究に従事．2021 年より富士通株式会社所属．

江田 智尊

2018 年九州大学大学院数理学府数理学専攻修了．同年より株式会社富士通研究所所属．サイバーセキュリティに関する研究に従事．2021 年より富士通株式会社所属．

大山 恵弘 （正会員）

2001 年東京大学大学院理学系研究科情報科学専攻修了．科学技術振興事業団研究員，東京大学大学院情報工学系研究科助手，電気通信大学大学院情報理工学研究科准教授を経て，2016 年より筑波大学システム情報系准教授．博士（理学）．システムソフトウェア，ソフトウェアセキュリティに関する研究に従事．