類似度に基づいた評価データの選別によるマルウェア検知精度の向上

村上純一† 鵜飼裕司†

†株式会社 FFRI
150-0013 東京都渋谷区恵比寿 1 丁目 18 番 18 号 東急不動産恵比寿ビル 4F
murakami@ffri.jp, ukai@ffri.jp

あらまし 近年，マルウェアの高度化が進んでおりパターンマッチング等の従来方式に基づいたマルウェア検知が困難になっている．新たな検知方式として機械学習を適用した手法が提案されており，従来に比べて高い検出率を実現できることが様々な研究により報告されている．一方でこれら機械学習による分類は，一般に学習データと傾向の異なる評価データについては著しく精度が下がることが知られている．そこで本研究では，評価データを学習データとの類似度に基づいて選別することで選別後の評価データに対して高い検出精度を向上させる手法について考察する．

Improving accuracy of malware detection by filtering evaluation dataset based on its similarity

Junichi Murakami† Yuji Ukai†

†FFRI, Inc.
4F TOKYU LAND CORPORATION EBISU Bld., 1-18-18 Ebisu, Shibuya, Tokyo, 150-0013, JAPAN
murakami@ffri.jp ukai@ffri.jp

Abstract In recent years, it has been getting more difficult to detect malware by a traditional method like pattern matching because of the improvement of malware. Therefore machine learning-based detection has been introduced and reported that it has achieved a high detection rate compared to a traditional method in various research. However, it is well-known that the accuracy of detection significantly degrades against data that differ from training dataset. This study provides a method to improve accuracy of detection by filtering evaluation dataset based on similarity between evaluation and training dataset.

1 はじめに

近年，標的型攻撃等に代表されるマルウェアを利用した企業，組織内の情報搾取を目的とした高度な攻撃が増加している．攻撃者は事前に市販のアンチウイルス製品にてこれらの攻撃

に利用するマルウェアが検知可能か確認し，検知不可能なマルウェアを利用することができるため，攻撃発生時点にてこれらを検知することは困難である．加えて，攻撃者はアンダーグラウンドマーケットにて流通している各種マルウェア作成キットを利用することで技術力を必要とせず，容易にマルウェア及びその亜種を作成することが可能である．

今日一般に広く普及しているパターンマッチング方式によるマルウェア検知は性質上，未知マルウェアを検知することは困難である．加えて前述の背景よりパターンデータの増加によるコンピュータリソースの逼迫と言う課題に直面している．そこで各アンチウイルスベンダーは，パターンデータをクライアントに配信しないレピュテーション方式，マルウェアの静的な情報及び挙動に基づいたヒューリスティック方式等の他方式に注力している．

中でも，昨今のビッグデータ活用の潮流から注目を集めているのが機械学習を応用したマルウェア検知である．機械学習の 1 タスクである分類(classification)によるマルウェア検知は，他の従来方式に比べて高い分類精度が確保できることが様々な研究により報告されている．一方で分類は，一般に学習データ及び評価データともに同じ分布の集合からサンプリングしたデータを利用して分類精度を評価することが前提となっており，学習データと評価データの傾向が大きく異なる場合，分類精度が著しく低下することが知られている（文献 1）．

そこで本研究では，まず実際のマルウェア及び正常系ソフトウェアにおけるこの傾向について調査し、結果を考察する．次に，これを踏まえて評価データと学習データの類似度を算出し，類似度の高い学習データが存在しない評価データをフィルタリングすることで分類精度を向上させる手法を提案，評価する．尚，本研究は分類に関する手法，乃至は特徴の選定等による分類精度の向上を検討するものではない．

2 関連研究

マルウェアに対する機械学習の応用例としては前述のように分類による検知が挙げられる．これらの研究は，学習及び評価データについてどのような特徴を抽出するかが主な要点である．採用する特徴については，大きく 2 つに大分される．一つは，ソフトウェアの静的な情報を特徴とするものであり，典型的な特徴としてソフトウェアのコード領域を逆アセンブルして得られた命令列の n-gram が挙げられる(文献 2)．他方は，マルウェアを実際に実行し，実行時の挙動を特徴として採用するものである(文献 3)． いずれの方式においても従来のアンチウイルス製品と比較して，高い検出率が得られることが報告されている．

また収集したマルウェア群を適切な種類，家系に分類，整理すると言った応用例も存在する．これは，機械学習における分類，クラスタリングと言う種類のタスクを利用したものである．単純にマルウェアを事前に用意したラベルに分類するケースもあれば，マルウェア群をクラスタリングすることで仲間外れに相当するデータを洗い出し，新種のマルウェアを発見すると言った応用例も存在する（文献 4，5）

3 本研究における前提

本研究は前述のように，「分類による分類精度は，学習データと評価データの傾向が大きく異なる場合に著しく低下する」という前提の下に成り立っている．そこで本節では，実際のマルウェア及び正常系ソフトウェアにおいて当該前提が適用可能か評価し，その結果について考察する．

3.1 学習及び評価データの用意

本評価を行うに当りマルウェアと正常系ソフトウェアの学習及び評価データを用意する必要がある．マルウェアについては，FFRI Dataset

2013 を利用し，正常系ソフトウェアについては独自にインターネット上から収集した計 3951 件のソフトウェアを動的解析し，FFRI Dataset 2013 と同様のデータを用意した．尚，これら正常系ソフトウェアについては，市販のアンチウイルス製品でスキャンを行い，マルウェアが含まれないことを確認した．どちらも動的解析の過程において他の実行ファイルを実行する等によりログファイル中に複数のプロセス挙動が含まれる場合がある．そのため，本研究では，ログファイル中の最初に記録されたプロセスを対象とした．

次に前述の前提を検証するため上記データを以下の要領でそれぞれ学習及び評価用に振り分けた．

(1) 全てのマルウェアについて相互の類似度を算出し，類似度が閾値を超えたもの同士をクラスタリングする．
(2) 上記クラスタにおいて属する要素が 2 以上の非単一要素クラスタの場合，その半分を学習用，残りを評価用にする．
(3) 単一要素クラスタについては全て評価用にする．
(4) 正常系ソフトウェアについても(1)～(3)の要領で学習及び評価データに振り分けを行う．

上記を実施するに当り，類似度の算出にオープンソースの機械学習フレームワークである Jubatus（文献 7）のレコメンデーション機能を利用した．Jubatus を利用した理由は，フレームワークに特徴ベクトル変換器が含まれており，データの前処理及び加工を自身で行う手間が省くことができ，効率的に評価を行うことができるためである．特徴ベクトルは，ログファイル中に含まれる対象プロセスの API ログに着目し，呼び出された API 名を単位とした n-gram の出現回数を採用した．例えば「A, B, C, D, A, B, C, E」と言った API コールシーケンスが存在した場合，「ABC 2 回」，「BCD 1 回」，「CDA 1 回」，「DAB 1 回」，「BCE 1 回」 と言う要領である(3-gram の場合)．API 名の n-gram を特徴として採用した理由は，プロセスの挙動と言う観点において他の情報に比べてより本質的な特徴になり得ると考えたためである．

レコメンデーションアルゴリズムには minhash（文献 8）を利用した．これは，Jubatus がサポートしているレコメンデーションアルゴリズムの中で比較的アルゴリズムに対するパラメータが少なく，非本質的な部分で迷うことなく利用できるためである．参考迄にレコメンデーションを行うプログラムである jubarecommender の設定ファイルを以下に示す．

```
{
        "method" : "minhash",
        "parameter" : {
          "hash_num" : 64
},

"converter" : {
        "string_filter_types": {},
        "string_filter_rules": [],
        "num_filter_types": {},
        "num_filter_rules": [],
        "string_types": {},
        "string_rules": [
          {"key":"*",    "except":"api_call",    "type":"space",
"global_weight":"idf", "sample_weight":"bin"},
          {"key":"api_call",                      "type":"space",
"global_weight": "idf", "sample_weight" : "log_tf"}
        ],
        "num_types": {},
        "num_rules": [{"key" : "*", "type" : "num"}]
      }
}
```

図 1 jubarecommender の設定ファイル

上記レコメンデーション機能により指定した 2 つの要素間の類似度を 0.0～1.0 の値で取得することができる．一例として，4-gram にて API 名の n-gram を利用した場合の結果を表 1に示す．尚，マルウェア，正常系ソフトウェアそれぞれについて呼び出された API 数が 4 に満たないものについては事前にデータより除外した．

表 1よりマルウェア，正常系ソフトウェアともに閾値が 1.0（完全一致）に近づくにつれ，クラスタ数が低下し，単一要素クラスタが増加していることが分かる．特徴的な点として閾値 0.85 から 0.9 の間での正常系ソフトウェアの変化が挙げられる．閾値 0.85 ではクラスタ数が 1532

件であったのに対し，閾値0.9では2513件と約1000 件増加しており，この増分の大半が単一要素クラスタであることが分かる．また，閾値1.0 では，正常系ソフトウェアは，全要素数の90%が単一要素クラスタであるのに対し，マルウェアは55%になっている．これは，マルウェアが正常系ソフトウェアに比べて多くの亜種によって構成されていることを示唆しており，FFRI Dataset 2013 中に含まれる亜種の構成を反映していると考えられる．一方で，今回用意したデータにおいては「正常系ソフトウェアはマルウェアに比べて類似度の高いデータが存在し難い」ことが伺える．

3.2　マルウェア分類及び結果

前述のマルウェア及び正常系ソフトウェアの学習及び評価データを利用してマルウェアの分類を行った．まず，マルウェア及び正常系ソフトウェアで学習を行い，次に非単一要素クラスタに相当するデータの評価を行った後，単一要素のデータの評価を行った．本検証にて期待される結果は，評価データ中の非単一要素クラスタについては高い分類精度が得られ，単一要素のデータについては，分類精度が低下するというものである．分類には類似度の算出同様，Jubatus の分類機能を利用した．分類アルゴリズムにはAROW（文献9）を利用し，特徴ベクトルについては類似度の計算同様 API 名の n-gram を利用した．AROW を利用した理由は，Jubatus がサポートしている分類アルゴリズムの中で他と比べて安定した高い分類精度が得られためである．参考迄に分類を行うプログラムである jubaclassifier の設定を以下に示す．

```
{
  "method" : "AROW",

  "converter" : {
    "string_filter_types": {},
    "string_filter_rules": [],
    "num_filter_types": {},
    "num_filter_rules": [],
    "string_types": {},
    "string_rules": [
      {"key":"*",    "except":"api_call",    "type":"space",
"global_weight":"idf", "sample_weight":"bin"},
      {"key":"api_call", "type":"space", "global_weight":
"idf", "sample_weight" : "bin"}
    ],
    "num_types": {},
    "num_rules": [{"key" : "*", "type" : "num"}]
  },

  "parameter" : {
    "regularization_weight": 1.0
  }
}
```

図 2 jubaclassifier の設定ファイル

4-gram での分類結果を表 2に示す．表 2より期待された通り，非単一要素クラスタであるマルウェアの TPR(True Positive Rate)が約98%であるのに対し，単一要素のマルウェアは81%となっており，17%の低下が確認できる．同様に，正常系ソフトウェアの FPR(False Positive Rate)に関しても非単一要素クラスタである要素については0.62%である一方，単一要素のものに関しては約 4.5%となっており，3.88%の FPR 上昇による精度悪化が確認できる．これにより，マルウェアにおいても前述の前提について一定の相関性があると考えられる．尚，Err に記載の数値は，データを評価するための学習結果が存在しない等の理由により評価できなかったデータである．

4　提案手法

本節では，前節の結果を踏まえて学習データに類似度の高いデータが存在しない場合，当該評価データを除外することで分類精度を高める手法について評価を行った結果を記載する。

表 1 マルウェア及び正常系ソフトウェアのクラスタリング結果

	閾値	総要素数	クラスタ数	最大クラスタの要素数	非単一要素クラスタ数	単一要素クラスタ数	総要素数に対する単一要素クラスタの割合(%)
マルウェア	0.8	2612	481	986	2247	365	14
	0.85		625	711	2130	482	18
	0.9		834	396	1950	662	25
	0.95		1155	193	1687	925	35
	1.0		1633	75	1184	1428	55
正常系	0.8	3951	652	2942	3392	559	14
	0.85		1532	1663	2619	1332	34
	0.9		2513	206	1722	2229	56
	0.95		3185	51	985	2966	75
	1.0		3636	24	397	3554	90

表 2 非単一要素及び単一要素クラスタの分類結果

	Total	TP	FN	TN	FP	Err	TPR	FPR	Precision
非単一要素	1814	978	20	796	5	15	0.97996	0.00624	0.99491
単一要素	2893	539	124	2127	100	3	0.81297	0.04490	0.84351

4.1 学習及び評価データの用意

前節同様，FFRI Dataset 2013 及び独自に収集した正常系ソフトウェア計 3951 件を対象とし，それぞれ無作為に二分割を行った．分割した一方をそれぞれ学習データ，他方を評価データとした．学習及び評価データの内訳を表 3に示す．

表 3 学習及び評価用データの内訳

	学習データ(件)	評価データ(件)
マルウェア	1302	1310
正常系	1988	1963

4.2 フィルタリング及び分類

表 3のデータについて3.2節同様，Jubatus を利用して分類による学習及び評価を行った．但し，今回はマルウェア及び正常系ソフトウェアについて学習後，事前に各評価データの学習データとの類似度を算出し，類似度が設定した閾値を超えないデータについては評価対象外とした．類似度の計算方法及び分類方法は3.1及び3.2節と同様である．

5 実験結果及び考察

表 4に分類結果を示す．各閾値において TP 及び FN の合計が評価対象となったマルウェアの件数であり，TN 及び FP の合計が評価対象となった正常系ソフトウェアの件数である．閾値 0 が類似度によるフィルタリングを行わず全ての評価データを分類した結果であり，閾値を 1.0 に向けて上昇するにつれ，評価対象とするデータの件数が減少していることが分かる．マルウェアは閾値 0.65，正常系ソフトウェアは閾値 0.7 で変化を見せており，TPR 及び FPR ともに精度が向上していることが確認できる．マルウェアにおいては閾値を上昇するにつれ，評価対象となるデータは亜種の集合となる．そのため，多くの亜種を持つマルウェアを検知し，それ以外についての FPR を低下させる用途においては本手法は有効な手段だと言える．

次に，閾値を上昇させた場合の評価対象となるデータ数の変化に着目する．閾値毎の全データ数に対するマルウェア，及び正常系ソフトウェアの評価対象数の割合を図 3に示す。

表 4 類似度に基づいたフィルタリングを適用した際の分類結果

閾値	Total	TP	FN	TN	FP	Err	TPR	FPR	Precision
0	3273	1202	103	1892	22	54	0.92107	0.01149	0.98203
0.6	3273	1202	103	1892	22	54	0.92107	0.01149	0.98203
0.65	3273	1202	103	1892	22	54	0.92107	0.01149	0.98203
0.7	3225	1175	100	1878	19	53	0.92157	0.01002	0.98409
0.75	3035	1082	85	1807	13	48	0.92716	0.00714	0.98813
0.8	2562	998	48	1470	5	41	0.95411	0.00339	0.99501
0.85	2012	931	29	1016	1	35	0.96979	0.00098	0.99893
0.9	1533	858	10	635	0	30	0.98848	0.00000	1.00000
0.95	1129	716	6	381	0	26	0.99169	0.00000	1.00000
1.0	652	469	1	160	0	22	0.99787	0.00000	1.00000

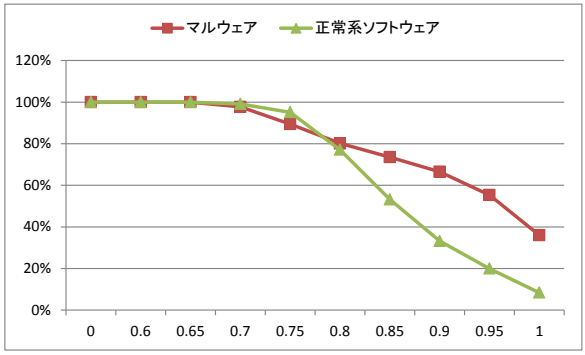

図 3 評価対象データ数の変化

図 3より，前述のように閾値が 1.0 に近づくにつれ評価対象となるデータの件数が減少しているが，マルウェアに比べて正常系ソフトウェアの方が，減少率が高いことが確認できる．これは前節で述べたように正常系ソフトウェアはマルウェアに比べて類似するデータが存在し難いことを考えると自明である．そのため本手法をアンチウイルス製品等の実際の利用シーンに適用した場合，亜種を多く持つマルウェアに対しては高い分類精度を確保できるものの閾値を上げていくにつれ，類似性の低いデータ，特に正常系ソフトウェアの多くが対象外になるものと考えられる．

次に，評価対象外としたデータの取扱いについて考察する．これらデータの取扱いは，下記の 3 つの対応が考えられる．

(1) 別の特徴を用いた分類を行う
本評価では，実行時に呼び出された API の n-gram を特徴としており，前述の評価結果はいずれも 4-gram での結果である．そのため，API 名以外の項目を特徴として採用し，学習・評価することでこれらデータについても正しく評価できる可能性がある．

(2) より多くのデータを収集し，学習させる
より多くのマルウェア及び正常系ソフトウェアを収集し，学習させることでこれらデータについても類似度の高いデータが出現し，将来的に非単一要素のクラスタを形成する可能性がある．

(3) 別の手法による検知を行う
機械学習の分類による検知をあきらめ全く別の手法による検知を行う．

(1)は，様々な手法が考えられるが本研究の対象外であるためここでは言及しない．(2)については，実効性についての課題が想定される．世の中に存在する全ての正常系ソフトウェアを収集することは現実的に困難であり，仮に収集することができたとしてもそれはマルウェアを含めた全てのソフトウェアと高い類似性を持つとは限らない．そのため現実世界においては常に学習データと傾向の異なる評価データに遭遇するリスクが存在し，この場合3.2節の評価結果より分類精度の低下は避けることができない．

本研究ではマルウェア及び正常系ソフトウェアともに限られたデータセットではあるが，その類似性の傾向が異なることが確認された．この場合，学習データと異なる傾向の評価データの出現による分類精度の低下は，特に正常系ソフトウェアにおいては本質的に避けることがで

きない可能性がある．そのため，(3)の一例として「マルウェアと正常系ソフトウェアを分類する」のではなく，「正常系ソフトウェアを検知する」技術の確立が今後の課題であると考えられる．

6　まとめ

本研究によりマルウェアには亜種が多く含まれているため正常系ソフトウェアとは類似性の面で傾向に違いがあることが確認された．また，マルウェア検知においても学習及び評価データ間で傾向が異なる場合，分類精度が低下することが確認された．これらを考慮し，類似度に基づいた評価データのフィルタリングを行った結果，分類精度の向上が確認されたが，フィルタリングの閾値とする類似度を上昇させるにつれ，評価対象となるデータ数が減少し，特に正常系ソフトウェアにおいてはこれが顕著であった．

そのため今後，マルウェアが現状のように亜種を多く含む形で増加していった場合，継続的に学習を行うことで機械学習によるマルウェア検知は一定の効果が見込めると考えられる．一方で，オリジナルとなる一意なマルウェアが増加していった場合，正常系ソフトウェア同様，学習及び評価データの傾向が異なることが想定され，分類精度が低下することは否めない．これについては，正常系ソフトウェアを検知する等，異なる技術の開発が課題であると考えられる．

参考文献

1) 森 信介：自然言語処理における分野適応 http://plata.ar.media.kyoto-u.ac.jp/mori/research/public/JSAI12Jul.pdf
2) Dragos Gavrilu Mihai Cimpoesu ¸Dan Anton, Liviu Ciortuz : Malware Detection Using Machine Learning
3) Yanfang Ye, Dingding Wang, Tao Li, Dongyi Ye：IMDS: Intelligent Malware Detection System
4) Philipp Trinius, Carsten Willems, Thorsten Holz, and Konrad Rieck：A Malware Instruction Set for Behavior-Based Analysis
5) 堀合啓一，今泉隆文，田中英彦：マルウェア亜種の動的挙動を利用した自動分類手法の提案と実装，情報処理学会論文誌，Vol. 50 No.4 1321-1333(Apr. 2009)
6) Konrad Rieck, Philipp Trinius, Carsten Willems, and Thorsten Holz: Automatic Analysis of Malware Behavior using Machine Learning
7) Jubatus: Distributed Online Machine LearningFramework, http://jubat.us/en/
8) Ping Li, Arnd Christian Konig, b-Bit Minwise Hashing, WWW, 2010
9) Koby Crammer, Alex Kulesza and Mark Dredze,Adaptive Regularization Of Weight Vectors,Advances in Neural Information Processing Systems, 2009
10) jubatus-example / malware classification, https://github.com/jubatus/jubatus-example/tree/master/malware_classification
11) 機械学習のセキュリティ技術応用, http://www.ffri.jp/assets/files/monthly_research/MR201306_Machine_learning_for_computer_security_JPN2.pdf