マルウェア感染端末の分類精度改善に向けたラベルなしネットワークログの活用法

西山 泰史[1]　熊谷 充敏[1]　神谷 和憲[1]　谷川 真樹[1]　高橋 健司[2]

概要：近年，機械学習を用いてマルウェア感染の有無を判定する手法が注目されている．しかし，学習に必要となるラベルのついた通信ログはセキュリティアナリストの分析が必要となるため，十分な量を収集するのは容易ではない．一方，ラベルのないログであれば，proxy ログや NetFlow など，ネットワーク運用で平常時から収集している大量のログを活用することができる．特に NetFlow は ISP などの大規模なネットワークからも収集可能であるため，世界規模のネットワークから Bot や C&C サーバなどマルウェアに関連した情報を取り込み，学習に活かすことができる．本稿では，ラベルのついたログとラベルのないログの両者を最大限活用するため，グラフベースの label spreading と教師あり logistic regression を組み合わせた半教師あり学習手法を提案する．本手法はモデルの解釈が可能かつ計算時間が現実的であるため，実運用に耐え得る手法である．本稿では，2 種類の実際の通信ログ（大企業の proxy ログ，大規模 ISP の NetFlow）を用いて，提案法と従来の教師あり学習の分類性能を比較し，提案法の優位性を示す．また，ラベルありの proxy ログだけを学習するだけでは検知できなかったが，提案法でラベルなしの NetFlow も学習に取り入れることによって検知できるようになったマルウェア感染端末についても考察する．

キーワード：感染検知，半教師あり学習，ログ分析，HTTP トラヒック，NetFlow

1. はじめに

マルウェアの感染被害を防ぐためには事前に感染を防ぐことが最善であり，ウイルス対策ソフトはその典型策である．しかし，攻撃者の手により次々と新種のマルウェアが作成されているため [1]，事前策のみで完全にマルウェア感染を防ぐことは難しい．そこで，ある程度の感染は避けられないものとして，ネットワーク内の通信ログを分析することで，マルウェア感染を早期に検知し，感染被害を最小限に抑える，という事後対策がとられている [2]．

実際に，多くのセキュリティベンダでは，SOC(Security Operation Center) と呼ばれる組織に専門性の高いセキュリティアナリストを配置して，顧客の通信ログを監視/分析するサービスを提供している [3]．しかし，全ての通信ログを人手で分析することは，人的リソースや金銭的コストの観点で難しい．そこで SOC では SIEM などのログ分析ツールに検知ルール（シグネチャ）を追加して，条件に合致したログのみを分析することで稼動を削減している．現状，この検知ルールは様々な情報ソースを用いてセキュリティアナリストが手動で追加しているが，攻撃者が次々に新種のマルウェアを作成しているため，すぐに陳腐化してしまう，という課題がある．

近年，機械学習を用いたログ分析ツール（以下，分類器と呼ぶ）が注目されている．新種のマルウェアは日々大量に作成されているものの，その多くは完全に新種ではなく，ソースコードの一部が再利用されていることが多い [4]．したがって，機械学習を用いれば，シグネチャで検知できないマルウェアでも，既知のマルウェアの通信パターンとの類似性を捉えて検知できる可能性がある．また，追加するシグネチャの数を削減できれば，セキュリティアナリストの負担軽減にもなる．ただし，一般に機械学習は検知精度を高めるため，大量のラベルあり学習用データを必要とする．ここで言うラベルあり学習用データとは解析済みのデータのことで，正規ユーザ由来の通信（良性）かマルウェア由来の通信（悪性）かがすでに判定されたログのことである．ラベルあり学習用データはセキュリティアナリストが手作業で詳細を分析して得る必要があるため，入手が難しい．一方，ラベルなし学習用データはラベル付けの必要がないため，大量に収集可能で，例えば SOC の未解析の proxy ログや ISP の未解析の NetFlow [5] が該当する．ラベルなし学習用データはラベルがないものの，通信の特徴

[1] 日本電信電話株式会社 セキュアプラットフォーム研究所
Nippon Telegraph and Telephone Corporation, Secure Platform Laboratories
[2] NTT Security

や関係性からマルウェアに関連した情報が得られる可能性がある．特に NetFlow は ISP などの大規模なネットワークからでも取得できるため，世界規模の情報の中から Bot や C&C サーバなどに関する情報が得られる期待がある．

本研究では，ラベルあり学習用データとラベルなし学習用データの両方の情報を最大限活用できる半教師あり学習を用いて，マルウェアに感染した疑いのある端末か否かを分類する機械学習分類器を作成する手法を考える．既存の半教師あり学習手法として，生成モデル（naive bayes [6]，deep generative models [7]），識別モデル（s3vm [8]，tsvm [9]），グラフベース（label propagation [10], label spreading [11]）などのアプローチが挙げられる．ただし，SOC の実運用においては，セキュリティアナリストが詳細を分析するため，分類理由が明確であること，計算時間が現実的であることが求められる．生成モデルはデータの生成過程のモデル化を行い，分類を行う手法であるが，そのモデルの正しさを検証することが難しく，分類理由は解釈しにくい．また，識別モデルは教師あり学習で用いられている手法をベースに，「データの分布密度が低い領域に分類境界が存在する」などの分類境界に関する仮定を置いて，分類を行う手法である．多くの場合，識別モデルは高精度となるが，その仮定は必ずしも正しいとは言えず，分類理由は不明確である．以上の理由から，本研究では，グラフベースの半教師あり学習の label spreading [11] と教師ありの logistic regression [12] を組み合わせた手法を提案する．詳細は 3.2 節にて記すが，提案法は特徴量ごとの重みが算出できるため，分類理由が明確で，計算時間も現実的な手法である．

提案法の有効性を確認するため，2 種類の実際の通信ログ（大企業の proxy ログ，大規模 ISP の NetFlow）を用いて，提案法と従来の教師あり学習の分類性能を比較する．2 章でシステムの概要，3 章で提案法の詳細，4 章で実験の概要および結果を示す．また，実験の中で，従来の教師あり学習では検知できなかったが，提案法で検知できたマルウェア感染端末について考察した結果を 5 章で述べる．

本論文の主な貢献は以下の 4 つである．

- 感染端末検知において，分類理由が解釈可能かつ計算時間が現実的な半教師あり学習手法を提案した．
- ラベルありの proxy ログとラベルなしの NetFlow のように，異なる種類のネットワークログに対して半教師あり学習を適用した場合について初めて検証し，その効果を示した．
- 2 種類の実網の通信ログに対して提案法と従来の教師あり学習を適用し，AUC と誤検知率が 0.1%となるよう閾値を調整した際の検知率で性能の比較を行い，提案法の優位性を示した．
- 従来の教師あり学習では検知できなかったが，提案法で新しく検知できるようになったマルウェア感染端末の実例を挙げ，検知理由を考察した．

2. システム概要

本研究の最終的な目標は，ラベルあり学習用データとラベルなし学習用データの両方を最大限活用して，端末が感染していそうか否か，つまり良性か悪性かを判定する機械学習の分類器を作成することである．SOC の実運用においては，分類器で分析対象を絞り込み，感染が疑われる（悪性）と判定された通信ログのみをセキュリティアナリストが詳細に分析することで運用される．

本稿では，以下の 2 ケースに対して提案法と従来の教師あり学習を適用し，分類性能を比較した．

[Case 1] ラベルあり学習用データ：proxy ログ
ラベルなし学習用データ：proxy ログ
テストデータ：proxy ログ

[Case 2] ラベルあり学習用データ：proxy ログ
ラベルなし学習用データ：NetFlow
テストデータ：proxy ログ

proxy ログはプロキシサーバから得られるログで，URL，送信元/宛先 IP アドレス，送信元/宛先ポート番号，ユーザエージェント，HTTP ステータスコード，タイムスタンプなどの情報が得られる．proxy ログは 1 ログあたりの情報量が多いため，SOC のログ分析サービスでは主に proxy ログを用いている．ただし，proxy ログは，機器の設定が必要である，情報量が多いため大規模なネットワークでは記録しづらい，詳細な個人情報を含む，などの理由から，様々な環境から収集することが難しい．

NetFlow はネットワーク内のルーティング装置を経由するパケットの統計情報を示す IOS 機能の 1 つで [5]，送信元/宛先 IP アドレス，送信元/宛先ポート番号，バイト数，タイムスタンプなどが記録されている．NetFlow は 1flow あたりの情報量が少ないため，詳細な分析には向かないが，収集しやすいというメリットがあり，ISP などの大規模なネットワーク網からも取得が可能である．

Case 1/Case 2 は共に SOC で使用する分類器を想定しており，実運用では，セキュリティアナリストが解析した後のデータをラベルあり学習用データ，分類したい顧客のログをテストデータに入力すればよい．Case 1/Case 2 それぞれのラベルなし学習用データには，セキュリティアナリストが解析する前の未解析の proxy ログ/未解析の NetFlow を入力することを想定している．Case 2 のように，大規模なネットワークから収集できる NetFlow をラベルなし学習用データとして用いることができれば，世界規模のトラヒックの中から有用な情報が得られる可能性がある．

なお，Case 1 において，本稿ではラベルなし学習用データとテストデータは異なるデータセットを用いている．同じデータセットを用いる場合 (トランスダクティブ学習)，新しいテストデータを判定する度に再学習が必要となるが，これは計算コストの観点で実運用では不適である．

3. システム詳細

3.1 機械学習

本節では提案法に用いた機械学習について記す．以下では簡単化のため 2 クラス分類の場合について述べる．

Label spreading: label spreading はグラフベースの半教師あり学習で，特徴量間の類似度を計算して，似た特徴量をもつデータに同じラベルを割り当てるアルゴリズムである [11]．元々のラベルありデータのラベル変更を許容する点が大きな特徴である．l 個の組 $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_l, y_l)\}$ をラベルありデータ，u 個の組 $\{\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, \cdots, \mathbf{x}_{l+u}\}$ をラベルなしデータと定義する．ここで，$\mathbf{x}_p \in \mathbb{R}^m$ はデータ点 p の m 次元の特徴ベクトルで，$y_p \in \{0, 1\}$ はラベルとする．

ここで，F_{ij} はデータ点 $\mathbf{x}_i$ のラベルが $j-1$ である確率を意味しており，$F \in \mathbb{R}_+^{n\times 2}$ $(n = l+u)$ は (i, j) 成分が F_{ij} であるベクトル関数行列であるとする．また，$Y \in \mathbb{R}_+^{n\times 2}$ は初期ラベル行列で，$\mathbf{x}_i$ のラベルが $y_i = j-1$ であれば $Y_{ij} = 1$，それ以外は $Y_{ij} = 0$ と定義する．label spreading のアルゴリズムは以下のようになる．

STEP Ⅰ　アファイン行列 $W \in \mathbb{R}^{n\times n}$ を

$$\begin{aligned} &i \neq j \qquad W_{ij} = \sigma \exp(-||\mathbf{x}_i - \mathbf{x}_j||^2), \\ &i = j \qquad W_{ij} = 0, \end{aligned} \tag{1}$$

と定義する．ただし，$\sigma \in \mathbb{R}_+$ はハイパーパラメータ．

STEP Ⅱ　(i, i) 成分が W の i 行と等しい対角行列 D を用いて，$S = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ と S を定義する．

STEP Ⅲ　$F(t+1) = \rho S F(t) + (1-\rho) Y$ という計算を F が収束するまで繰り返す．t は繰り返しの回数を意味しており，初期値は 0 である．ρ は閉区間 $[0, 1]$ で定義されている．

STEP Ⅳ　$F^* = \lim_{t\to\infty} F(t)$ と定義する．このとき，各 $\mathbf{x}_i$ は $y_i = \mathrm{argmax}_{j\in\{1,2\}} F^*_{ij}$ とラベル付けされる．

STEP Ⅰでは，$\mathbf{x}_i$ と $\mathbf{x}_j$ 間の特徴量の類似度を計算している．$i \neq j$ の場合，$\mathbf{x}_i$ が $\mathbf{x}_j$ に近いほど W_{ij} は大きくなる．STEP Ⅱでは正規化を行っている．STEP Ⅲではラベル情報の伝搬を行っている．なお，ρ はクランピング係数で初期ラベルの変更のしやすさを調整するパラメータである．

Logistic regression: 2 クラスを $y \in \{0, 1\}$ としたとき，これらの事後確率は以下のように書ける [12]．

$$p(y = 1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x}), \tag{2}$$

$$p(y = 0|\mathbf{x}) = 1 - \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x}), \tag{3}$$

$$\sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x}) = \{1 + \exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x})\}^{-1}. \tag{4}$$

ただし，$\mathbf{w} = [w_1, \cdots, w_{m'}] \in \mathbb{R}^{m'}$ は m' 次元の重みベクトル，$\mathbf{x} \in \mathbb{R}^{m'}$ は m' 次元の特徴ベクトルである．また，T は転置を意味する．2 クラス分類においては，ある閾値 T_h を決めて，$p(y = 1|\mathbf{x}) > T_h$ となる場合は $\mathbf{x}$ に 1，$p(y = 1|\mathbf{x}) \leq T_h$ の時は 0 のラベルを割り当てればよい．

このとき，L_2 正則化付きの誤差関数は以下のようになる．

$$\begin{aligned} E = -C \sum_{n'=1}^{N'} \{&(1 - y_{n'}) \log(1 - \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x}_{n'})) \\ &+ y_{n'} \log \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x}_{n'})\} + \frac{1}{2}||\mathbf{w}||^2. \end{aligned} \tag{5}$$

ここで，$\{\mathbf{x}_{n'}, y_{n'}\}_{n'=1}^{N'}$ はラベルあり学習用データ，$C \in \mathbb{R}_+$ はハイパーパラメータである．$w_k > 0 (1 \leq k \leq m')$ となる重みは $y = 1$ のクラスに，$w_k < 0$ となる重みは $y = 0$ のクラスに分類されるのに寄与する．また，w_k の絶対値が大きいほどクラス分類に大きく寄与している．つまり，この重みベクトル w_k を見ることで，どの特徴量がクラス分類結果に大きく寄与したかを調べることができる．

3.2 提案法

本節では，提案法の詳細について述べる．提案法の手順は以下のようになる．

STEP 1　ラベルあり学習用データ $D_L = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_l, y_l)\}$ とラベルなし学習用データ $D_U = \{\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, \cdots, \mathbf{x}_{l+u}\}$ を用意し，良性 = 0，悪性 = 1 となるよう，ラベル $\{y_1, y_2, \cdots, y_l\}$ を付与する．

STEP 2　ラベルあり/ラベルなし学習用データ D_L/D_U に label spreading を適用し，良性/悪性となる確率を意味する $F^* \in \mathbb{R}_+^{n\times 2}$ を算出する．

STEP 3　誤ってラベル付けしたデータを学習させることを避けるため，閾値 T_m と T_l を導入する $(0 \leq T_l \leq T_m \leq 1)$．これらの閾値はハイパーパラメータである．STEP 2 のラベルなし学習用データ D_U のうち，悪性となる確率が T_m 以上のものに悪性のラベル，悪性となる確率が T_l 以下のものに良性のラベルを割り当てる．ラベルを割り当てたラベルなし学習用データの集合を D_{U_n} と定義する．

STEP 4　ラベルあり学習用データ D_L とラベルを付与したラベルなし学習用データ D_{U_n} を結合して新しいラベルあり学習用データ $D_{new} = D_L \cup D_{U_n}$ を作る．

STEP 5　新しいラベルあり学習用データ D_{new} に対して logistic regression を適用し，分類器を作る．

STEP 6　STEP 5 で作成した分類器を用いて，テストデータが良性か悪性かを分類する．

STEP 3 の閾値 T_m と T_l は，良性か悪性かを分類しにくいラベルなし学習用データを logistic regression の学習に取り込むことを回避するために導入している．もし誤って

ラベル付けしたデータを元に分類器を作ってしまうと分類精度が劣化してしまうためである．T_m と T_l を厳しめに設定すると（つまり T_m を 1 に近い値，T_l を 0 に近い値にする），STEP 5 で学習に取り入れる，ラベルを付与したラベルなし学習用データが減少するため，新規の悪性情報を学習に取り入れる可能性が減ってしまうが，誤ってラベル付けしたデータを学習に取り込む可能性を減らすことができる．逆に T_m と T_l を甘めに設定すると（つまり T_m と T_l を 0.5 に近い値にする），誤ってラベル付けしたデータを学習に取り込む可能性が上がるが，新規の悪性情報を学習に取り入れる可能性があがる．なお，STEP 2 と STEP 5 では異なる特徴量を使用することができる点に注意されたい．本稿で用いた特徴量については 4.2 節にて言及する．

SOC の実運用においては，最終的にセキュリティアナリストが詳細を分析するため，分類理由が明確であること，計算時間が現実的であることが求められる．提案法は STEP 5 で logistic regression を用いているため，特徴量の重みベクトル $\mathbf{w}$ を算出すれば，どの特徴量が良性/悪性判断に寄与しているかを解釈することができる．また，STEP 2 の label spreading は，(1) 式の W_{ij} を算出すると，ラベル付けしたラベルなし学習用データに最も近いラベルあり学習用データを探すことができるため，ラベルなし学習用データになぜ良性/悪性のラベルが割り当てられたかが解釈できる．したがって提案法は分類理由が明確な手法であると言える．また，label spreading と logistic regression は一般に他の非線形な機械学習モデル（deep learning や support vector machine など）に比べて計算コストが小さい．計算時間の実験結果については 4.4 節にて述べる．以上より，提案法は実運用上有効な手法であると考えられる．

4. 評価実験

4.1 データセット

本節では比較実験で用いた通信ログについて記す．

proxy ログは 2 つの異なるデータソースから得ている．良性のデータは，十分なセキュリティマネジメントが行われている，ある大企業 1 社の社内網から取得した．また，悪性のデータはマルウェアの検体を VirusTotal [13] からダウンロードし，それをサンドボックスシステム [14] を用いて動的解析して得た．各検体は各々 SHA1 Hash が異なり，ランサムウェア，アドウェアなど様々な種類を含んでいる．また，全ての検体は VirusTotal の中の複数のアンチウイルスソフトから悪性と判定されたもので，各種ウイルス対策ソフトによる検知傾向がなるべくランダムになるよう，日々最新のものを一定数収集して動的解析している．なお，1 検体が 1 端末のログに対応している．

NetFlow は，アメリカ，ヨーロッパ，日本など世界規模にサービスを提供している実在の大規模 ISP から 1:10000 のサンプリングをかけて収集した．ただし，NetFlow は双方向のトラヒックログであるため，送信元と宛先が逆になって記録されているログが存在する．もし，送信元と宛先を区別せずに学習に取り入れてしまうと精度が悪化する可能性があるため，well-known ポート（0-1023）が宛先側，high ポート（1024-65535）が送信元側となるよう一部のデータセットの方向を入れ替えた．なお，well-known ポートから well-known ポートへの通信や，high ポートから high ポートへの通信は対象とせず，本実験では除去した．

表 1 作成したデータセットのホスト数とログ数

	Name	Type	Label	# of hosts	# of logs
Labeled Training	L20	Proxy	Legitimate	20	787
			Malicious	20	419
	L100	Proxy	Legitimate	100	3,766
			Malicious	100	2,240
	L500	Proxy	Legitimate	500	18,969
			Malicious	500	11,575
	L2500	Proxy	Legitimate	2,500	100,229
			Malicious	2,500	60,340
Unlabeled Training	UP200	Proxy	-	200	6,515
	UP2000	Proxy	-	2,000	67,643
	UP20000	Proxy	-	20,000	648,565
	UN10k	NetFlow	-	10,000	212,462
	UN50k	NetFlow	-	50,000	1,222,605
	UN250k	NetFlow	-	250,000	5,608,194
Validation	V	Proxy	Legitimate	1,000	44,175
			Malicious	1,000	18,800
Test	T1	Proxy	Legitimate	1,000	44,375
			Malicious	1,000	19,046
	T2	Proxy	Legitimate	1,000	44,262
			Malicious	1,000	19,537

実験で用いたデータセットの端末数とログ数を表 1 として示す．これらのデータセットは (L20) ⊂ (L100) ⊂ (L500) ⊂ (L2500)，(UP200) ⊂ (UP2000) ⊂ (UP20000)，(UN10k) ⊂ (UN50k) ⊂ (UN250k) となるよう作成した．また，極力取得日が古いものから順に，ラベルあり学習用データ，ラベルなし学習用データ，検証用データ，テストデータとなるようデータセットを作成している．

なお，検証用データセットはホールドアウト検証を行うために用意している．ハイパーパラメータを調整する際に，一般的には交差検定などが用いられているが，交差検定を用いると，時系列的に後に発見されたマルウェアの検体のログを用いて分類器を生成し，前に発見されたマルウェアを検知する，という検証を行ってしまう．しかし，本研究の目的は新しいマルウェアをどれだけ誤検知少なく検知できるのかを検証することであるため，本稿では時系列を考慮してホールドアウト検証を採用した．また，Case 1 の proxy ログのラベルなし学習用データは良性と悪性の端末数を 1:1 にして混ぜ，ラベル情報を削除して作成した．

表 2　各特徴量の候補の平均 AUC

Feature	Mean AUC
Destination IP Address*	**0.9017**
Destination Port Number*	0.5342
HTTP Method*	0.5331
HTTP User Agent*	**0.9604**
Domain Name*	**0.9983**
URL Path*	**0.9972**
URL Path Element*	**0.9980**
URL Parameters*	0.5005
URL Query*	0.8649
Top Level Domain*	0.7481
Length of Domain Name**	**0.9315**
Length of URL**	**0.9490**
Length of URL Path**	**0.9394**
Length of URL Query**	0.8847

4.2　特徴抽出

本節では実験に用いた特徴量について記す．proxy ログの特徴量の候補としては広く一般に用いられているものを用いた [15, 16]．特徴量の候補は表 2 の通りである．表中の URL path element は URL パスを “/” で区切った各単語を意味している．例えば，“http://www.corporation.co.jp/RD/index.php” であれば，“RD” と “index.php’ が URL path element に該当する．なお，特徴ベクトルを作成する際には，各端末ごとに 1 つの特徴ベクトルを作成した．表 2 中で * がついた特徴量の候補は bag-of-words モデル [17] を用いて，** がついた特徴量の候補は学習用データに出現し得る最大の値で割り算して正規化し（例えば，URL の長さの場合は 2083），特徴ベクトル化した．

表 2 にある特徴量の候補は広く使われている特徴量ではあるが，これら全ての特徴量が良性/悪性判定に有効であるとは限らない．もしあまり有効でない特徴量を入れて学習してしまうと，過学習や計算時間の増大が生じうる可能性がある．そこで，各特徴量の候補がどの程度判定結果に寄与しているかを調べ，効果の高い特徴量の候補のみを特徴量として採用した．本研究では，特徴量を決定するため，L_2 正則化付きの logistic regression を各特徴量の候補それぞれに適用して AUC(Area Under the Curve) を算出し，AUC が高いものを特徴量として採用した．なお，AUC とは 0 から 1 の値をとる分類器の性能を表す指標で，1 に近いほど分類性能が良いとされている．

特徴量を選択する際には，L2500 を学習用データ，V をパラメータチューニング用のデータ，T1 と T2 をテストデータとして用いた．表 2 は T1 と T2 の平均 AUC を記載している．なお，logistic regression のハイパーパラメータは，$C \in \{10^A\}_{A=-5}^{5}$ の範囲で決定した．

表 2 の結果から，Case 1 の STEP 2 と 5 では，太字の候補を特徴量として採用した．ただし，Case 2 の場合は，NetFlow は HTTP や URL に関する情報がないので，STEP 2 では宛先 IP アドレスと宛先ポート番号を特徴量として label spreading を適用し，STEP 5 では表 2 の太字の候補に宛先ポート番号を加えて特徴量とした．なお，Case 2 の STEP 5 で NetFlow に logistic regression を適用する際には，情報がない項目は 0 として特徴ベクトル化した．

表 3　実験の種類

Experiment	Labeled Training	Unlabeled Training
EXPT 1	L20	UP200 or UP2000 or UP20000
EXPT 2	L100	UP200 or UP2000 or UP20000
EXPT 3	L500	UP200 or UP2000 or UP20000
EXPT 4	L2500	UP200 or UP2000 or UP20000
EXPT 5	L20	UN10k or UN50k or UN250k
EXPT 6	L100	UN10k or UN50k or UN250k
EXPT 7	L500	UN10k or UN50k or UN250k
EXPT 8	L2500	UN10k or UN50k or UN250k

4.3　実験概要

提案法の効果を示すため，本稿では Case 1 と Case 2 の 2 パターンについて，提案法と代表的な教師あり学習の性能比較を行った．代表的な教師あり学習法としては，logistic regression (LR)，Gini impurity で算出した random forest (RF)，RBF カーネルの support vector machine (SVM) を用いた．なお，RF は実験結果が毎回変動するため，5 回同じ実験を繰り返し，その平均値を実験結果として採用した．

評価指標としては，AUC と $\mathrm{TPR}_{\mathrm{FPR=0.1\%}}$ を用いた．$\mathrm{TPR}_{\mathrm{FPR=0.1\%}}$ は，false positive rate (FPR) が 0.1% 以下となるよう閾値を調整した際の true positive rate (TPR) と定義している．実際のログ分析サービスでは，セキュリティアナリストの稼動を削減するために，誤検知率（FPR）が低いときの検知率（TPR）が重要となる．

本稿では，表 3 の 8 つの実験設定の各々に対して，6 パターンの機械学習方法を適用した．例えば，EXPT 1 では，L20 に LR/RF/SVM を適用した場合，L20 と UP200/UP2000/UP20000 に提案法を適用した場合の計 6 パターンの実験を行った．なお，各実験においては，V を用いて AUC が最大となるようハイパーパラメータを調整し，T1 と T2 を用いてテストした際の AUC/$\mathrm{TPR}_{\mathrm{FPR=0.1\%}}$ の平均値を比較した．なお，提案法のハイパーパラメータは全部で 5 種類あり，$\sigma \in \{1, 10, 20, \cdots, 90, 100\}$，$\rho \in \{0.9, 0.99\}$，$C \in \{10^A\}_{A=-5}^{5}$，$T_m \in \{0.9, 0.99, 0.999\}$，$T_l \in \{0.1, 0.01, 0.001\}$ の範囲から選択した．また，教師あり LR の L_2 正則化のパラメータを $\{10^A\}_{A=-5}^{5}$，RF の木の数を $\{10, 100, 1000\}$，SVM の rbf カーネルのパラメータを $\{1, 5, 10, 50, 100, 500, 1000\}$，SVM の L_2 正則化のパラメータを $\{10^{-4}, 5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}\}$ の範囲から決定した．

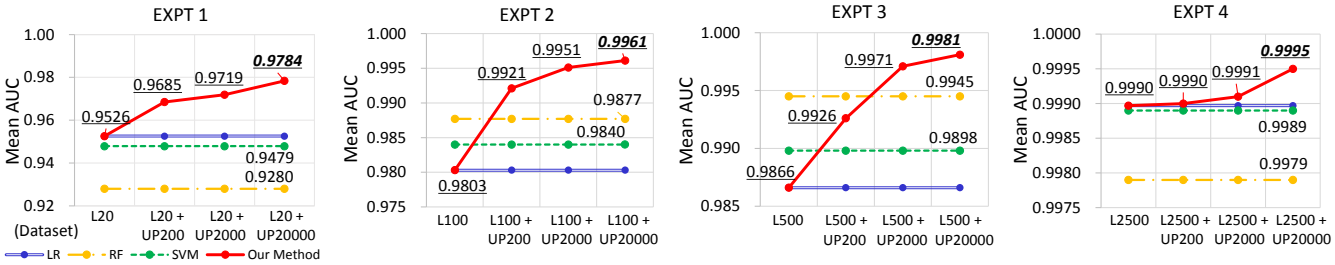

図 1　Case 1：平均 AUC (ラベルあり proxy ログ＋ラベルなし proxy ログ)

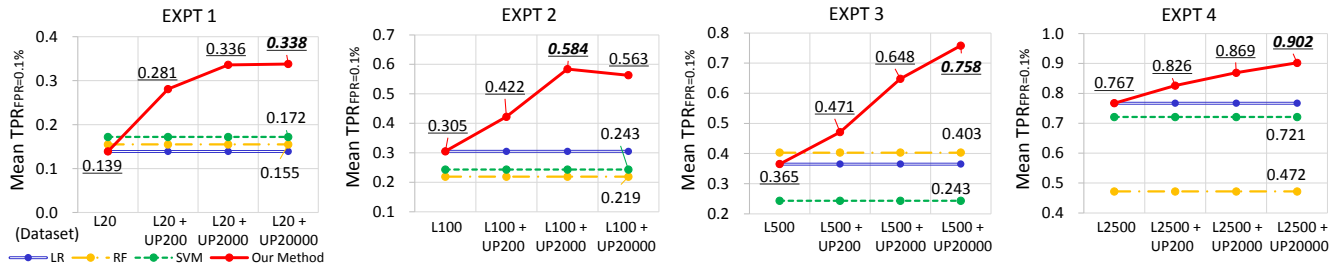

図 2　Case 1：平均 $\mathrm{TPR}_{\mathrm{FPR=0.1\%}}$（ラベルあり proxy ログ＋ラベルなし proxy ログ)

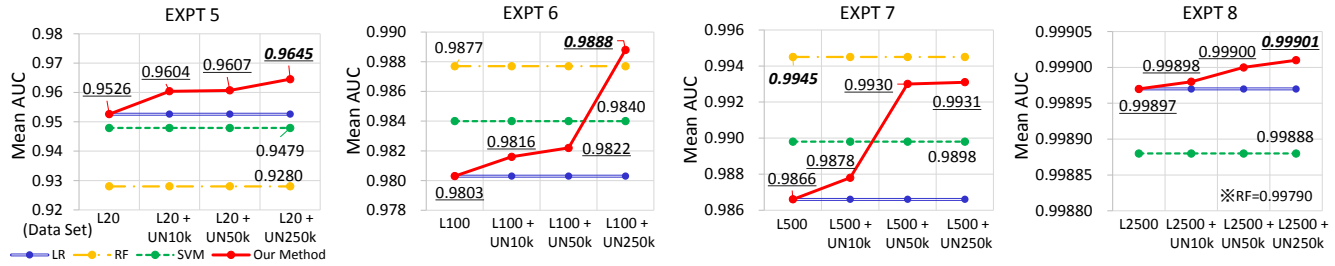

図 3　Case 2：平均 AUC（ラベルあり proxy ログ＋ラベルなし NetFlow）

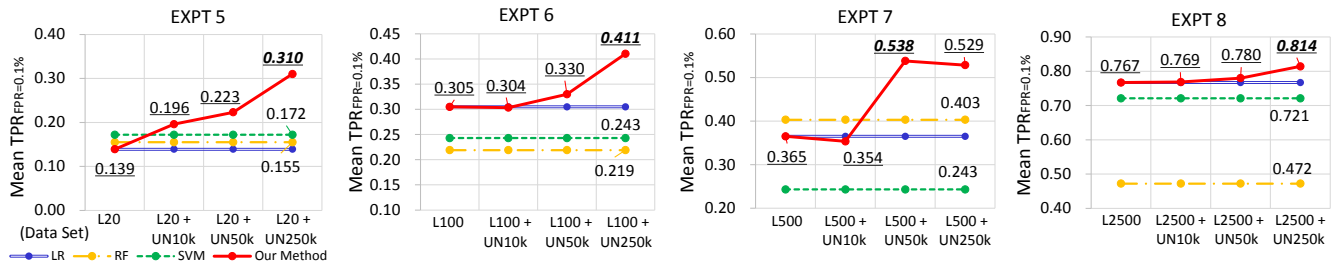

図 4　Case 2：平均 $\mathrm{TPR}_{\mathrm{FPR=0.1\%}}$（ラベルあり proxy ログ＋ラベルなし NetFlow）

4.4　実験結果

図 1，図 2 は EXPT 1 ～ 4（Case 1）の実験結果であり，テストデータとして T1 と T2 を用いた場合の AUC および $\mathrm{TPR}_{\mathrm{FPR=0.1\%}}$ の平均値を示している．横軸は実験に用いた学習用のデータセットを意味している．教師あり学習はラベルなしデータを学習に利用できないため，LR，RF，SVM の実験結果は横軸に水平となっている．図 1 と図 2 から，ラベルあり学習用データの数を固定した場合，提案法は概ねラベルなし学習用データが増加するにつれて精度が向上しており，UP20000 から学習した場合の提案法が最も高精度であることが読み取れる．一般に，誤ってラベル付けしたデータを学習に取り込んでしまうと分類精度が悪化する可能性があるが，提案法は十分にラベルあり/なし学習用データの両方から学習できたと言える．

同様に，図 3，図 4 は EXPT 5 ～ 8 （Case 2）の実験結果である．ラベルあり学習用データの数を固定した場合，提案法はラベルなし学習用データが増加するにつれて精度が向上しており，UN250k から学習した場合の提案法が概ね最も高精度であることが読み取れる．NetFlow の 1flow あたりの情報量は proxy ログに比べて少ないので，Case 1 ほどの精度の向上は見られなかったが，提案法は proxy ログベースの分類器にラベルなしの NetFlow を学習させた場合でも精度を向上させることができた．

表 4 は全プロセスの計算時間を表している．なお，実験

表 4　平均計算時間 (second)

Experiment	Supervised Learning			Our Method		
	SVM	RF	LR	UP200	UP2000	UP20000
EXPT 1	22	22	12	21	37	682
EXPT 2	27	41	13	21	43	707
EXPT 3	185	156	21	30	54	738
EXPT 4	10716	2061	69	93	121	1989
Experiment	Supervised Learning			Our Method		
	SVM	RF	LR	UN10k	UN50k	UN250k
EXPT 5	22	22	12	51	308	1886
EXPT 6	27	41	13	61	395	2238
EXPT 7	185	156	21	88	689	3460
EXPT 8	10716	2061	69	148	845	5107

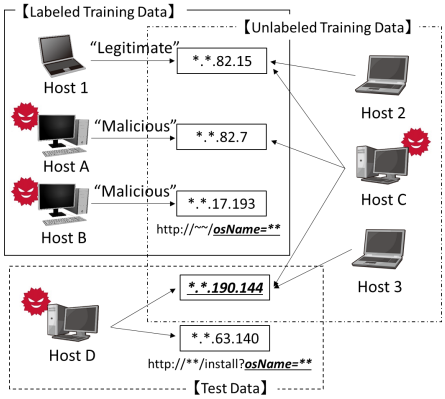

図 5　Case 2 で新しく検知できたマルウェア感染端末の例

環境は以下の通りである．

- CPU: Intel(R) Xeon(R) CPU E5-2660 v3 2.60GHz
- OS: Ubuntu 14.04
- Framework: Python, scikit-learn

表 4 から，提案法は教師あり学習に比べて最大 50 倍程度のデータを学習しているにも関わらず，現実的な計算時間に収まっていることが読み取れる．

なお，紙面の都合上省略するが，本実験では学習用データセットの良性/悪性端末の比率を 1:1 に固定したが，不均衡な場合でも同様の結果が確認できた．

5. 考察

本章では，ラベルあり学習用データからだけでは検知できなかったが，我々の手法で検知することができた新しいマルウェアの感染端末の例について述べる．図 5 は EXPT 8 (Case 2) において，ラベルありの proxy ログとラベルなしの NetFlow の両方を用いることで新しく検知できるようになったマルウェア感染端末の例である．Host 1 ～ 3 は良性もしくは良性と予想されるホストを意味しており，Host A ～ D は悪性もしくは悪性と予想されるホストを意味している．図 5 の新しいマルウェアの感染端末 Host D は proxy ログのラベルあり学習用データ（L2500）に現れる特徴量をあまり含んでいなかったため，ラベルあり学習用データだけでは検知されなかった．しかし，ラベルなし学習用データとして用いた NetFlow（UN250k）の中に，テストデータと同じ特徴量（宛先 IP アドレス）を含んでいたため，我々の手法では検知できたと考えられる．

次に，なぜ提案法がこのマルウェアの通信を検知することができたのかをより詳細に考察する．まず，Host A は感染端末でその通信先の IP アドレス*.*.82.7 は悪性の通信先であるという情報がラベルあり学習用データ（L2500）で与えられていた．つまり，*.*.82.7 は VirusTotal に使用されている複数のアンチウイルス製品でマルウェアと判定された検体の通信先である．また，NetFlow のラベルなし学習用データ（UN250k）の中で Host C が*.*.82.7 と*.*.190.144 に接続している，という情報が与えられていた．このとき，Host A と Host C は一部似た特徴ベクトルになるので，提案法の STEP 2 で Host C を特徴ベクトル化して label spreading を適用すると，Host C は感染の疑いがある端末として悪性に近い F^* が付与される．また，STEP 5 で logistic regression を適用すると IP アドレス*.*.190.144 は感染の疑いがある通信先として学習される．

さらにラベルあり学習用データ（L2500）の中で，Host B が感染端末でその通信先*.*.17.193 への通信ログの URL のパス部分に osName=**（**部分には実際の OS 名が記載されていたが，セキュリティ上の理由でマスキングしている）という文字列が含まれていた．これに提案法を適用すると，osName=**という文字列は悪性に多い文字列であると学習される．実際，提案法で使用した logistic regression の重みを見ると，*.*.190.144 や osName=**は正の値，つまり悪性に寄与しやすい特徴量となっていた．これら 2 つのスコアが積み重なった結果，提案法では Host D が感染の疑いがある端末と判定されるようになった．

実際に，Host D から発された通信ログを VirusTotal を用いて分析したところ，*.*.190.144 や*.*.63.140 への通信は複数のベンダの製品がマルウェアだと判定した検体の通信ログの一部であることがわかった．ただし，*.*.190.144 に関する pcap を分析したところ，*.*.190.144 は完全に悪性の IP アドレスというわけではなく，良性/悪性両方のコンテンツを提供するダウンロードサイトであった．実際，Host 3 のような正規ユーザと思われる複数のホストが*.*.190.144 に通信を行っていた．したがって，Host D は NetFlow もしくは proxy ログのいずれか片方だけでは良性か悪性かを判定することは難しい．一方で我々の手法は NetFlow の IP アドレス情報と proxy ログの URL の文字列の情報の両方を統合できるので，Host D を悪性と判定することができるようになったと考えられる．

なお，Host C と通信をする宛先 IP アドレスの全てが悪性であるとは限らない点には注意が必要である．図 5 の*.*.82.15 は正規の宛先 IP アドレスの例である．*.*.82.15

は感染している疑いのある端末 Host C と通信しているが，もしこの*.*.82.15 がある程度著名な良性のウェブサイト（例：Google，Facebook，twitter など）である場合は，ラベルあり学習用データ（Host 1）やラベルなし学習用データ（Host 2）の中に*.*.82.15 と通信をしているログが含まれる可能性がある．このような場合，全体的なログの関係性を統計的に考慮して，logistic regression の重みは負，つまり良性に寄与しやすい特徴として学習される．

6. 関連研究

半教師あり学習を用いたログ分析の関連研究について述べる．Shi ら [18] は proxy ログに対して，グラフベースの半教師あり学習と RF を独立に適用して，未知の悪性ドメインを検知する手法を提案している．半教師あり学習部分では，ドメインのみを用いて通信関係を示す二部グラフを作成し，悪性ドメイン情報の拡張を行っている．ただし，重み付けのない二部グラフを用いているので，複数の特徴量の場合を扱うことができない．提案法は複数の特徴量を用いることができるため，より精度の高い分類を行うことができる．これは，表 2 や図 1 ～ 4 からも明らかである．

Beaugnon ら [19] はアクティブラーニングを用いた侵入検知法について述べている．アクティブラーニングは半教師あり学習の一種で，効率的に分類性能を高めてくれそうなラベルなしデータを人に推薦し，人にラベルを付けてもらったデータを追加学習する，という操作を繰り返す機械学習法である．一方で，本稿の手法はラベルなしデータへのラベル付けを必要としない．NetFlow のように 1flow あたりの情報量は少ないが膨大な量のデータを扱う際には，ラベル付与の難しさや量の問題から，本稿のような追加のラベル付けを必要としない手法が優位であると考えられる．

7. 結論

本稿では，label spreading と logistic regression を組み合わせた半教師あり学習を用いて，マルウェア感染の有無を判定する分類器の作成法を提案し，実在の大企業網の proxy ログや大規模 ISP の NetFlow を用いて，提案法の優位性を示した．提案法はモデルの解釈が可能かつ計算コストが低い手法であるため，SOC での実運用上有効な手法である．また，モデルが解釈できる長所を活用して，従来の教師あり学習では検知できなかったが，提案法で検知できたマルウェア感染端末についても考察し，その効果を示した．提案法は異なる種類のログから学習した場合においても効果が確認できたため，大規模 ISP の NetFlow のように世界規模のログからマルウェアに関する知見を取り込み，学習に活かすことができると考えられる．

参考文献

[1] AV-Test, https://www.av-test.org/en/statistics/.

[2] Bartos, K. and Sofka, M. (2016). Optimized invariant representation of network traffic for detecting unseen malware variants, In Proceedings of the 25th USENIX Security Symposium, 807/822.

[3] McAfee, Creating and maintaining a SOC, https://www.mcafee.com /sg/resources/white-papers/foundstone/wp-creating-maintaining-soc.pdf.

[4] Jang, J., et al. (2011). BitShred: feature hashing malware for scalable triage and semantic analysis, In Proceedings of the 18th ACM Conference on Computer and Communications (CCS), 309/320.

[5] Claise, B. (2004). Cisco systems NetFlow services export version 9, https://tools.ietf.org/html/rfc3954.

[6] Nigam, K., et al. (2000). Text classification from labeled and unlabeled documents using EM, Machine Learning-Special issue on information retrieval, Vol.39, 103/134.

[7] Kingma, D., et al. (2014). Semi-supervised learning with deep generative models, Advances in Neural Information Processing Systems (NIPS), 3581/3589.

[8] Bennett, K. and Demiriz, A. (1999). Semi-supervised support vector machines, Advances in Neural Information Processing Systems (NIPS), 368/374.

[9] Joachims, T. (1999). Transductive inference for text classification using support vector machines, In Proceedings of the 16th International Conference on Machine Learning (ICML), 200/209.

[10] Zhu, X., et al. (2003). Semi-supervised learning using gaussian fields and harmonic functions, In Proceedings of the 20th International Conference on Machine Learning (ICML), 912/919.

[11] Zhou, D., et al. (2004). Learning with local and global consistency, Advances in Neural Information Processing Systems (NIPS), 321/328.

[12] Bishop, C. (2006). Pattern recognition and machine learning, Springer-Verlag New York Inc.

[13] Virustotal, https://www.virustotal.com/.

[14] Aoki, L., et al. (2011). Controlling malware http communications in dynamic analysis system using search engine, In Proceedings of 3rd IEEE International Workshop on Cyberspace Safety and Security (CSS), 1/6.

[15] Ma, J., et al. (2011). Learning to detect malicious URLs, ACM Transactions on Intelligent Systems and Technology (TIST), Vol.2 Issue3, No.30.

[16] Sahoo, D., et al. (2017). Malicious url detection using machine learning: a survey, arXiv preprint arXiv:1701.07179.

[17] Zhang, Y., et al. (2010). Understanding bag-of-words model: a statistical framework, International Journal of Machine Learning and Cybernetics, Vol.1, 43/52.

[18] Shi, L., et al. (2015). A hybrid learning from multi-behavior for malicious domain detection on enterprise network, In Proceedings of the 2015 IEEE International Conference on Data Mining Workshop (ICDMW), 987/996.

[19] Beaugnon, A., et al. (2017). Ilab: an interactive labeling strategy for intrusion detection, In Proceedings of the 20th International Symposium on Research in Attacks, Intrusions and Defenses (RAID), 120/140.