# 効果的なネットワークインシデント検知のための半教師ありデータスクリーニング

村井 光[1,3,a)]　正代 隆義[2,3,b)]

**概要**：ネットワーク・インシデントを早期に検知するために，統計や可視化等の技術によるインシデントの機械的発見技法の開発が盛んに行われている．どのような発見技法でも，既知のマルウェアによる単純なスキャンやバックスキャッタなどが新しい攻撃を隠してしまい，本来のインシデント発見に支障を来すという共通の問題がある．自動化されたスクリーニングは，この問題の解決策として有望な技術のひとつである．ここで，スクリーニングとは，設定ミスや既知のマルウェアによる単純なパケットを削除し，新しい攻撃を発見しやすくすることである．本論文では，Blum と Chawla(2001) により提案されたグラフベースの半教師あり学習を用いたスクリーニング手法を提案し，その有効性を検証するためにダークネット観測網に到達したデータに対する実験結果を報告する．

**キーワード**：半教師あり学習，最小カット，データスクリーニング，インシデント検知，ダークネット観測

# A Semi-Supervised Data Screening Method for Network Traffic Data

Hikaru Murai[1,3,a)]　Takayoshi Shoudai[2,3,b)]

***Abstract:*** To construct efficient countermeasures against critical incidents in Internet, many effective systems for early detection of the tends of such incidents are being investigated in many projects. One of the nasty problems in such systems is the existence of illegal but simple packets caused by well-known malwares, because those packets make anomaly detection harder. To remove such accesses by ordinary malware from the results of network monitoring, we propose an automatic data screening method by using the graph-based semi-supervised learning method (Blum and Chawla, 2001). Besides, we show its effectiveness by experimental results on the darknet traffic data.

***Keywords:*** Semi-supervised learning, Mincut, Data screening, Incident detection, Darknet monitoring

## 1. はじめに

近年，コンピュータウイルスに起因する情報漏えい等，ネットワーク上でのインシデント (悪質な攻撃) は日増しに問題となってきている．とりわけ，ハーダーと呼ばれる悪意のある攻撃者が，複数の善意のコンピュータを遠隔操作可能にして，DOS 攻撃やスパムメール送信等の様々な脅威を発生させる，いわゆるボットネットに対して，その悪質性が深刻であることから，早期に発見し駆除する技術の開発が早急に求められている．

このような突発するネットワークインシデントを常時監

[1] 九州大学大学院システム情報科学府, 福岡市西区元岡 744
Department of Informatics, Kyushu University, Motooka 744, Nishi-Ku, Fukuoka, 819-0395, JAPAN
[2] 九州国際大学国際関係学部, 北九州市八幡東区平野 1-6-1
Faculty of International Studies, Kyushu International University, Hirano 1-6-1, Yahata-Higashi-ku, Kitakyushu, 805-8512, JAPAN
[3] 九州先端科学技術研究所, 福岡市早良区百道浜 2-1-22
Institute of Systems, Information Technologies and Nanotechnologies (ISIT), Momochihama 2-1-22, Sawara-ku, Fukuoka, 814-0001, JAPAN
[a)] sc109058@gmail.com
[b)] shoudai@isb.kiu.ac.jp

視する仕組みとして，ダークネットの観測がある．ダークネットとは，インターネット上で到達可能な IP アドレスのうち，特定のホストコンピュータが割り当てられていない（使われていない）アドレス空間のことである．ダークネットには，正常なパケットが届くことは稀である．それにも関わらず，ダークネットに届くすべてのパケットについて新しい悪意のある攻撃によるものか，そうではないかを判定することは非常に難しく，時間のかかる仕事である．なぜなら，ダークネットに到達するパケット数は，使われていない IP アドレスであるにも関わらず膨大な数であり，これからも増加していくと考えられているためである．例えば，日本のある組織に設置された匿名ダークネットに 2013 年において観測されたパケット数は約 128.8 億にのぼる．

多くの研究が，統計や可視化等の技術を用いて新しい攻撃の検知を目標にしている．一方でほとんどの手法に共通の課題が存在する．それは，古くから存在するマルウェアが行う IP アドレス空間のスキャンやポートスキャン，バックスキャッタ等，新しい攻撃を隠してしまうトラフィックが多数に存在していることである．このようなボットネットと比べれば悪質とは言えない既知のマルウェアによるトラフィックをスクリーニングして，深刻な被害を及ぼすマルウェアによるトラフィックを発見しやすくすることは，トラフィック数が急増している近年において，悪質なマルウェア早期発見のための重要な前処理となっている．

トラフィック解析におけるスクリーニングの必要性について以下のようなものがあげられる．

- ノイズを削除することによる早期検出等の精度上昇
- 計算時間の削減

本研究では，これらの目的を達成するために半教師あり学習 [8] によるトラフィックデータのスクリーニング手法を提案し，その評価を行う．従来から行われている最も基本的なスクリーニング手法として，特定のポートや SYN/ACK フラグを持つパケットを全て削除する方法がある．これは，例えば，特に監視する必要がない SYN/ACK フラグや RST フラグを持つパケットを削除することは特に問題はない．しかし，特定のポートに到達するパケットを全て削除してしまうと攻撃の手がかりとなりうる必要なデータも同時に削除してしまう可能性があり，特に慎重に行わなければならない．次に，フィルタリングと呼ばれる手法がある．現在までに判明しているワームやマルウェアの特徴を登録し，それらに対応するパケットを削除するという方法である．フィルタリング手法の欠点には，ウイルスの亜種に対応できない点や登録するデータが多くなりすぎる点があげられる．近年では，亜種が誰でも簡単に作成できてしまうため亜種の進化のスピードに登録が追いつかないことも挙げられる．

機械学習によるスクリーニングには，Tsuruta ら [7], Okamoto と Shoudai[5] による頻出系列スクリーニングがあげられる．この頻出系列スクリーニングは，パケットの $IP$ アドレス，ポートの情報を利用して，頻出するパターンを被覆率や文字列パタンのサイズを考慮し削除するという手法である．この頻出系列スクリーニングを適用することで短時間に大量到達するパケット群（本論文ではこれをスパイクと呼ぶ）を自動で除去できることがわかっている．ただし，この頻出系列スクリーニングの課題として，用いている情報が少ないことや発見されにくい攻撃（10 分間に 1 度などのスロースキャン攻撃）を捕らえることが難しい点である．これらのことをまとめて，従来のトラフィックデータに対するスクリーニングには次のような課題がある．

- 用いる情報を増やすことによる表現力の向上
- 攻撃とは関係のないトラフィックだけ削除する正確性
- 新規の攻撃や亜種に対する適応性

これらの課題を達成するために，本論文では，最小カットを用いた半教師あり学習によるトラフィックデータ・スクリーニング手法を提案する．最小カットを用いた半教師あり学習は，Blum と Chawla[1] によって提案されたグラフベースの半教師あり学習 [6] の一つである．本論文の手法では，正負のラベル有りパケットとラベル無しパケットから重み付き有向グラフを構成し，それに対して最小カットによる半教師あり学習を実施することで，ほとんど全てのパケットに対してラベル付けを行う．そして，ラベル付けされたパケットのうち，削除して構わないと判断されたパケットを削除する．本論文では，さらに，提案したスクリーニング手法をダークネットに到達したトラフィックデータに適用した実験結果を報告し，本スクリーニング手法のトラフィックデータに対する有効性を実証する．

## 2. 準備

### 2.1 ダークネットにおけるトラフィック解析

ダークネットは，インターネットから到達可能な未使用の IP アドレス群を意味する．通常の通信は実在する相手に対してパケットを送信するため，ダークネットに向けてトラフィックが発生することはない．従って，ダークネット宛に送信されるパケットは，通常の通信ではなく，何らかの不正な活動に起因するものである可能性が高い．いくつかの国内外の機関では，こうしたダークネット宛のパケットを観測することで，インターネット上の不正な活動の傾向を把握する試みが行われている．

ダークネットに到達するトラフィックデータ中のパケットには以下のようなものが含まれている．

- マルウェアによるスキャン
  - ワーム型マルウェア (自身を複製し拡散するマルウェア) の探索活動
  - マルウェア感染の大局的活動
  - 感染爆発の予兆
- DDOS 攻撃の跳ね返り

- 設定ミス
- リフレクション攻撃の準備活動

また，本論文で扱うダークネットのトラフィックデータには次のような情報が含まれている．

(1) IP アドレス
(2) ポート番号
(3) プロトコルを示すフラグ (TCP, UDP)
(4) 時刻
(5) シーケンス番号
(6) ACK 番号

トラフィックデータを解析することは，データに内在するインシデントを明らかにする過去の状況把握だけでなく，近いうちに深刻な打撃を起こしそうな攻撃の早期検知ができると考えられている．しかし，ダークネットに到達したトラフィックデータでさえ，その量は巨大であり，手作業だけで解析するのは不可能である．そのため，トラフィックデータを解析するシステムや技術の開発が集中的に行われている．

### 2.2 半教師あり学習

半教師あり学習とは，正負のラベル有りデータとラベル無しデータの両方を利用して，教師付き学習や教師なし学習よりもよい結果を出すために用いられる．半教師あり学習が注目される動機として，適用可能なデータが多く得られ，かつ精度の向上が期待できることがあげられる．実際，現実世界で得られるデータでは，ラベル有りデータが少ない一方で，ラベル無しデータは簡単に，大量に得られることが多い．今回の実験データとして用いるダークネットのトラフィックデータも，悪意のあるトラフィックであるものには，ラベルが付けられているものがあるが，その数は少ないため半教師あり学習を行うことによって悪意のあるトラフィックを捕らえる精度をあげることを目標としている．

半教師あり学習には，分類器に基づく手法やデータに基づく手法が提案されている．本研究ではグラフベースの手法，具体的には，データから重み付き有向グラフを構成し，それに対して最小カットを計算することでラベル付けを行う手法 [1] を用いる．一般的に，半教師あり学習は次のように定義される．$X$ をデータの全体とし，$Y$ をラベルの集合とする．本論文でのラベルは 1 (positive) と 0 (negative) の 2 種類である．すなわち，$Y = \{0, 1\}$ である．入力は，$X$ の 2 つの部分集合 $L = \{x_i \mid 1 \leq i \leq \ell\}$ と $U = \{x_i \mid \ell + 1 \leq i \leq \ell + u\}$ である．$L$ の各要素には，$Y$ のラベルが割り当てられている．すなわち，前もって定義された関数 $f_L : L \to Y$ が与えられている．$L$ をトレーニングデータと呼ぶ．また，文脈より混乱が無い場合は，$U$ をラベル無しデータと呼ぶ．出力は，$L \cup U$ の各要素に $Y$ のラベルを対応させる関数 $f : L \cup U \to Y$ で，$f$ を $L$ に制限した関数 $f|L$ が $f_L$ に一致するものである．この関数を様々な指標に基づいて評価する．

半教師あり学習のうち分類器に基づく手法とは，初期分類器から始め，反復的に分類器を洗練する手法である．これには Self-training や Co-training がある [8]．これらは，初期に間違えると間違いが増幅するため，ラベル有りデータの信頼度が全体の信頼度を決定する特徴がある．また，トラフィックデータには様々なウィルスの亜種も含まれ，さらにその出現速度も速いため，同一のウィルスの素性だけでうまくラベル付けできるとは限らないので，分類器を反復的に用いる手法では，精度の良いラベル付けが期待しづらい．一方，本論文で用いたグラフベースの手法では，類似データは同一ラベルを持つ傾向があるという仮定から，データ間の類似度を定義・グラフ化し，そのグラフに対して，グラフベースのアルゴリズムを用いてラベル付けする．このことより，グラフ理論などの数学的な背景が確立することに加え，よいグラフが得られていればよい性能が期待できる．逆に言えば，適切なグラフ構造や有向辺の重みを定義することが，その性能に大きく影響を与える．以降では，トラフィックデータを適切な重み付きグラフ構造として定義する方法を提案し，最小カットによる半教師あり学習によりスクリーニング手法を提案する．

## 3. グラフベースの半教師ありスクリーニング

本章では，グラフベースの半教師あり学習 [1] を用いたトラフィックデータのスクリーニング手法について述べる．

### 3.1 トラフィックデータ上の距離

グラフを用いた半教師付き学習には頂点と辺が必要になる．トラフィックデータの 1 パケットをグラフ上の 1 頂点とみなし，頂点間の類似度を辺で示す．ここでの頂点間の類似度は，トラフィックデータから得られる情報（宛先 IP アドレス，送信元 IP アドレス，宛先ポート，送信元ポート，プロトコルフラグ（TCP, ICMP, UDP など），シーケンス番号，TTL，ACK 番号）を用いて定める．具体的には各情報に重みを決め，各頂点間の重さを計算し，$k$-近傍法により最も近い 3 つの頂点に対して辺を作る．これにより，似ている頂点間に辺ができるため類似性の高い頂点を，同一のクラスタに分類することが可能となる．

前述したように，本手法ではトラフィックデータの 1 パケットをグラフ上の 1 頂点とみなす．まず，このグラフに対して，頂点間の距離を定義する．

**IP ヘッダ**

IP ヘッダとは，パケットの先頭に付加される情報で，この情報を用いて送り先や経路を決定する．IP ヘッダには，始点 IP アドレス，終点 IP アドレス，TTL 値 (Time to Live, 生存時間) などの全プロトコル共通の情報が含まれている．類似度グラフを作成するにあたって，IP ヘッダから

は，始点 IP アドレス，TTL 値，識別番号から距離を定義する．

- 始点 IP アドレスに関する距離の定義
  トラフィックデータ $x_1$ と $x_2$ の始点 IP アドレスをオクテット毎に分割し，その値を先頭からそれぞれ $a_{1,1}, a_{1,2}, a_{1,3}, a_{1,4}$ および $a_{2,1}, a_{2,2}, a_{2,3}, a_{2,4}$ とし，$x_1$ と $x_2$ の始点 IP アドレスにより生じる距離 $d_{adr}(x_1, x_2)$ を次のように定める．

$$d_{adr}(x_1,x_2) = \begin{cases} 4 & \text{if } a_{1,1} \neq a_{2,1}, \\ 3 & \text{if } (a_{1,1} = a_{2,1}) \wedge (a_{1,2} \neq a_{2,2}), \\ 2 & \text{if } (a_{1,1} = a_{2,1}) \wedge (a_{1,2} = a_{2,2}) \\ & \quad \wedge (a_{1,3} \neq a_{2,3}), \\ 1 & (a_{1,1} = a_{2,1}) \wedge (a_{2,1} = a_{2,2}) \\ & \quad \wedge (a_{1,3} = a_{2,3}) \wedge (a_{1,4} \neq a_{2,4}), \\ 0 & \text{otherwise.} \end{cases}$$

  始点 IP アドレスによる類似度はオクテッド毎に異なるよう定義する．具体的には，上位オクテッドの違いを下位オクテッドより重く距離づけする．これは IP アドレスの機関毎の割り当て基準に合わせたものである．
- TTL による距離の定義
  TTL (Time to Live) とは，パケットの生存期間を意味し，送信元が指定するパケットが経由するルータの上限の値である．ルータを 1 つ経由するたびに値が 1 ずつ減り，0 になるとパケットは破棄される．この機能は，パケットが設定ミスによりネットワーク内を無限に巡回することを避ける効果がある．OS やそのヴァージョンにより，TTL の初期設定値が異なっている．そのため TTL の値を距離の要素として用いることで，特定の OS が原因となる攻撃を捕らえることを目標としている．具体的には，TTL と代表的な OS やバージョンには次のような関係がある．

| OS | TTL 初期値 |
|---|---|
| Mac OS X | 64 |
| Windows 98 | 128 |
| Windows XP | 128 |
| MPE/IX (HP) | 200 |
| OpenBSD | 255 |

  Eto ら [2] は，到達したパケットの TTL から初期値 TTL を推測する基準を示し，それが妥当であることを示した．到達したパケットの TTL 値 (左列) から推測される初期 TTL 値 (右列) を表 1 にあげる．

| 観測時 TTL 値 $d$ | 推測初期 TTL 値 |
|---|---|
| $0 \leq d \leq 21$ | 32 |
| $22 \leq d \leq 39$ | 48 |
| $40 \leq d \leq 59$ | 64 |
| $60 \leq d \leq 89$ | 100 |
| $90 \leq d \leq 120$ | 128 |
| $120 \leq d \leq 159$ | 168 |
| $160 \leq d \leq 189$ | 200 |
| $190 \leq d \leq 255$ | 255 |

**表 1** 観測時 TTL 値により推測される初期 TTL 値 ([2])

  $x_1$ と $x_2$ の観測時 TTL 値により推測される初期 TTL 値を $a_{ttl}(x_1), a_{ttl}(x_2)$ とする．このとき，TTL 値から得られる距離 $d_{ttl}(x_1, x_2)$ を次のように定める．

$$d_{ttl}(x_1,x_2) = \begin{cases} 2 & \text{if } a_{ttl}(x_1) \neq a_{ttl}(x_2), \\ 0 & \text{otherwise.} \end{cases}$$

- 識別番号による距離の定義
  大きなデータを運ぶ時，ネットワークの境界で，一度に送信できる最大のデータ量 (MTU) のサイズを超えてしまう場合がある．このデータを送るには，複数のパケットに分割して送る必要がある．その際，もとは同じで分割されたデータなのか，全く別のデータなのかを識別するために識別番号が使用される．分割されたデータの IP ヘッダ内の識別番号には全て同じ値が入るので，識別番号をパケットの類似度の一部とするのは，至極妥当であると考える．
  $x_1$ と $x_2$ の識別番号を $a_{id}(x_1), a_{id}(x_2)$ とする．このとき，識別番号から得られる距離 $d_{id}(x_1, x_2)$ を次のように定める．

$$d_{id}(x_1,x_2) = \begin{cases} 1 & \text{if } a_{id}(x_1) \neq a_{id}(x_2), \\ 0 & \text{otherwise.} \end{cases}$$

以上より，$x_1$ と $x_2$ の IP ヘッダ情報により生じる距離 $d_{ip}(x_1, x_2)$ を次のように定義する．

$$d_{ip}(x_1,x_2) = d_{adr}(x_1,x_2) + d_{ttl}(x_1,x_2) + d_{id}(x_1,x_2).$$

**TCP プロトコル**

TCP プロトコルには，その TCP ヘッダに始点ポート番号，終点ポート番号，シーケンス番号，確認応答番号が含まれる．この情報を用いて，通信先の確認 (3 way handshake) をしてコネクションの確立やデータの欠損確認，アプリケーションへの仲介 (ポート番号指定) などを行う．

- 始点・終点ポート番号に関する距離の定義
  TCP プロトコルであるパケット $x_i$ $(i = 1, 2)$ に含まれる始点ポート番号と終点ポート番号をそれぞれ $sp_i$, $dp_i$ $(i = 1, 2)$ とする．$x_1$ と $x_2$ の始点ポートにより生じる距離を $d_{sport}(x_1, x_2)$ と定義し，終点ポート番号により生じる距離を $d_{dport}(x_1, x_2)$ と定義する．

$$d_{sport}(x_1,x_2) = \begin{cases} 2 & \text{if } sp_1 \neq sp_2, \\ 0 & \text{otherwise.} \end{cases}$$

$$d_{dport}(x_1, x_2) = \begin{cases} 4 & \text{if } dp_1 \neq dp_2, \\ 0 & \text{otherwise.} \end{cases}$$

過去の事例より，マルウェアは特定の終点ポートを狙って攻撃するパタンが多い．例えば，後章で述べるMorto（モルト）と呼ばれるマルウェアは，Windowsのリモートデスクトップ接続を悪用して拡散するマルウェアで，主に終点ポート番号 3389 をターゲットに攻撃を行う．これらのような事例を考慮し，終点ポート番号の違いを始点ポート番号の違いより重要視している．

- シーケンス番号に関する距離の定義
  シーケンス番号とは，TCP 通信の際につけられる通し番号で，パケットの正しい順番や途中の欠落を確認するために用いる．送信するデータ 1 バイト毎に 1 ずつ昇順に割り当てどこまでデータを送信したのかを指定する．ただし，初期値が 0 ではなくランダムな値に設定される．パケット $x_1$ と $x_2$ に含まれるシーケンス番号の値をそれぞれ $seq_1$, $seq_2$ とし，$x_1$ と $x_2$ のシーケンス番号により生じる距離を $d_{seq}(x_1, x_2)$ と次のように定義する．

$$d_{seq}(x_1, x_2) = \begin{cases} 2 & \text{if } |seq_1 - seq_2| > 4 \\ 0 & \text{otherwise.} \end{cases}$$

- 確認応答番号に関する距離の定義
  確認応答番号とは，受信したデータに対して，どこのバイト位置までを受信したかを表す ACK 番号は，データを受信した側が，どこまで受信したかを示すために用いる．受信したデータのシーケンス番号に対応しており，受信が完了したデータ位置のシーケンス番号に 1 を加えた値を返すことになっている．パケット $x_1$ と $x_2$ に含まれる確認応答番号の値をそれぞれ $ack_1$, $ack_2$ とし，$x_1$ と $x_2$ の確認応答番号により生じる距離を $d_{ack}(x_1, x_2)$ と次のように定義する．

$$d_{ack}(x_1, x_2) = \begin{cases} 2 & \text{if } |ack_1 - ack_2| > 4 \\ 0 & \text{otherwise.} \end{cases}$$

以上より，パケット $x_1$ と $x_2$ がともに TCP プロトコルによる通信のとき，$x_1$ と $x_2$ の TCP プロトコルにより生じる距離 $d_{tcp}(x_1, x_2)$ を次のように定義する．

$$\begin{aligned} d_{tcp}(x_1, x_2) = {} & d_{sport}(x_1, x_2) + d_{dport}(x_1, x_2) \\ & + d_{seq}(x_1, x_2) + d_{ack}(x_1, x_2). \end{aligned}$$

**UDP プロトコル**

次に UDP プロトコルについて説明する．UDP は TCP と比較してデータサイズがかなり小さい．これは，シーケンス番号や確認応答番号などの情報が含まれていないためである．UDP プロトコルに含まれる情報は，始点ポート番号，終点ポート番号，UDP データの長さのみである．これによりデータ処理速度が速くなる．始点・終点ポート番号については，TCP プロトコルと同じ役割である．UDP データの長さとは，ヘッダとデータを含むデータグラム全体のバイト数のことである．

- UDP プロトコルに関する距離の定義
  パケット $x_1$ と $x_2$ がともに UDP プロトコルによる通信のとき，$x_1$ と $x_2$ の UDP プロトコルにより生じる距離 $d_{udp}(x_1, x_2)$ を次のように定義する．

$$\begin{aligned} d_{udp}(x_1, x_2) = {} & d_{sport}(x_1, x_2) + d_{dport}(x_1, x_2) \\ & + d_{udp\_len}(x_1, x_2). \end{aligned}$$

ここで，$d_{sport}(x_1, x_2)$ と $d_{dport}(x_1, x_2)$ はそれぞれ，始点ポート番号と終点ポート番号によって生じる距離で，TCP プロトコルと同様に定義される．また，UDP パケットの長さによって生じる距離 $d_{udp\_len}(x_1, x_2)$ は次のように定義する．パケット $x_1$ と $x_2$ の UDP データの長さをそれぞれ，$len_1$ と $len_2$ で表す．

$$d_{len}(x_1, x_2) = \begin{cases} 1 & \text{if } len_1 \neq len_2, \\ 0 & \text{otherwise.} \end{cases}$$

**ICMP プロトコル**

最後に，$ICMP$ プロトコルについて説明する．$ICMP$ とは，エラー通知や制御メッセージを転送するためのプロトコルで，コンピュータ間の通信状態を確認するために用いられる．$ICMP$ プロトコルには，タイプ，コード，データの情報がある．本手法では，始点・終点ポート番号の他にコードを距離の対象とする．

- ICMP プロトコルに関する距離の定義
  パケット $x_1$ と $x_2$ がともに ICMP プロトコルによる通信のとき，$x_1$ と $x_2$ の ICMP プロトコルにより生じる距離 $d_{icmp}(x_1, x_2)$ を次のように定義する．

$$\begin{aligned} d_{icmp}(x_1, x_2) = {} & d_{sport}(x_1, x_2) + d_{dport}(x_1, x_2) \\ & + d_{icmp\_code}(x_1, x_2). \end{aligned}$$

ここで，$d_{sport}(x_1, x_2)$ と $d_{dport}(x_1, x_2)$ はそれぞれ，始点ポート番号と終点ポート番号によって生じる距離で，TCP プロトコルと同様に定義される．また，ICMP パケットのコードによって生じる距離 $d_{icmp\_code}(x_1, x_2)$ は次のように定義する．パケット $x_1$ と $x_2$ のコードをそれぞれ，$code_1$ と $code_2$ で表す．

$$d_{code}(x_1, x_2) = \begin{cases} 1 & \text{if } code_1 \neq code_2, \\ 0 & \text{otherwise.} \end{cases}$$

パケット間の距離

以上のことより，本論文では，2 つのパケット $x_1$ と $x_2$ 間の距離 $d(x_1, x_2)$ を次のように定義する．

$$d(x_1,x_2) = \begin{cases} d_{ip}(x_1,x_2) + d_{tcp}(x_1,x_2) \\ \quad \text{if } x_1 \text{ と } x_2 \text{ がともに TCP プロトコル,} \\ d_{ip}(x_1,x_2) + d_{udp}(x_1,x_2) \\ \quad \text{if } x_1 \text{ と } x_2 \text{ がともに UDP プロトコル,} \\ d_{ip}(x_1,x_2) + d_{icmp}(x_1,x_2) \\ \quad \text{if } x_1 \text{ と } x_2 \text{ がともに ICMP プロトコル,} \\ \infty \quad \text{otherwise.} \end{cases}$$

### 3.2 最小カットアルゴリズムによる半教師あり学習

Blum と Chawla[1] により提案された最小カットによるラベル付けアルゴリズムを，トラフィックデータのラベル付けに合わせて概要を述べる．$k$ はあらかじめ定められた 1 以上の整数である．本論文では，このアルゴリズムを MSSL (Mincut Semi-Supervised Learning) と呼ぶ．

アルゴリズム MSSL

$L$ を正負のラベル有りパケット集合，$U$ をラベル無しパケット集合とする．

(1) 任意の 2 パケット $x_1, x_2 \in L \cup U$ に対して，第 3.1 章で定義した $d(x_1, x_2)$ を計算する．

(2) 重み付き有向グラフ $G = (V, E)$ を次のように作成する．

(a) 頂点集合を $V = L \cup U \cup \{v_+, v_-\}$ とする．

(b) 任意の頂点 $x \in L \cup U$ に対して，$x$ からの異なる距離を小さい順に $k$ 番目まで求める．その距離を $D_1, \ldots, D_k$ としたとき，$x$ からの距離が $D_j$ $(1 \le j \le k)$ である全ての頂点 $x'$ に対して，有向辺 $(x, x')$ を作成し，その重みを $k - j + 1$ とする．

(c) 任意の正ラベル頂点 $x \in L$ に対して，有向辺 $(v_+, x)$ と $(x, v_+)$ を作成し，その重みを無限大とする．

(d) 任意の負ラベル頂点 $x \in L$ に対して，有向辺 $(v_-, x)$ と $(x, v_-)$ を作成し，その重みを無限大とする．

(e) 同一ラベルを持つ頂点間には，有向辺が無ければ，有向辺を作成し，その重みを無限大に更新する．

(3) 上記で構成した重み付き有向グラフに対して，最大流最小カットアルゴリズムを適用し，$v_+$ から $v_-$ に至る最大流と最小カットを計算する．

(4) 求められた最小カットを適用することによって $V$ を次の 3 つの集合に分割する

- $V_+$ : $v_+$ から到達可能な $V$ の頂点集合，
- $V_-$ : $v_-$ へ到達可能な $V$ の頂点集合，
- $V_0$ : $v_+$ からも $v_-$ への到達不能な $V$ の頂点集合．

(5) $V+$ に含まれる頂点に正ラベルを，$V_-$ に含まれる頂点に負ラベルを付ける．

最小カットによる半教師あり学習の精度については，$k = 1$ のとき，最小カットによって生じる *LOOCV error* の値は，最近傍に重さ 1 の辺をつける際の最小カットのコストに等しいという保証がある．*LOOCV error* とは 1 個抜き交差検証を行った際の誤りの数を意味する．

## 4. 半教師ありスクリーニング実験と評価

現在，世界中でネットワークインシデントの早期検知システムの開発が行われている．インシデントの早期検知システムの最大の目的は，既知のマルウェアの攻撃を検知し，攻撃発生の警告を出すことではなく，今までに認識されていない新しい攻撃パタンを検知し，早期に注意喚起を促すことである．そのため，既知のマルウェアの攻撃によるパケット群は既に対応済みという意味で，新しい攻撃パタン発見の妨げとなる．このような既に攻撃パタンがわかっているマルウェアのパケット群をあらかじめスクリーニングし，新しい攻撃パタン発見のサポートを行うことが本研究の主目的である．

トラフィックデータから既知のマルウェアの攻撃によるパケット群を除去する実験を行うために，既知のマルウェアの攻撃によるパケット群か否かを，正と負でラベル付けしたデータが必要である．正ラベル付けされたデータは既知のマルウェアによるものと判定され，負ラベルのものはそうではないと判定されたものとする．本実験では，ラベル有りデータとして，川村ら [3], [4] が開発した非負値行列分解 (Non-negative Matrix Factorization, NMF) によるネットワークインシデント早期検知システムによりラベル付けされたデータを用いた．川村らのインシデント早期検知手法を NMF エンジンと呼ぶ．

本論文では，与えられたラベル無しデータと NMF エンジンによるラベル有りデータに対して，アルゴリズム MSSL を適用して，新たに正ラベルと判定されたデータを削除するという手順でスクリーニングを行った．以降の章で，その実験結果を報告する．

### 4.1 2011 年 7 月の NICT ダークネットデータ

独立行政法人・情報通信研究機構 (NICT) では，インターネットトラフィックデータの挙動把握のためダークネットを設置し，監視している．本実験の最初のデータは，NICT のダークネットで 2011 年 7 月に観測されたトラフィックデータである．前処理として，どのマルウェア攻撃にも無関係な RST フラグ付きのパケットは削除されている．以降では，24 時間のデータを 30 分毎，計 48 個の時間帯に分けて取り扱った．この分割による $i$ 番目 $(i = 1, 2, \ldots, 48)$ の時間帯を第 $i$ 時間帯と呼ぶ．NICT のダークネットでは，30 分間に数万パケットの単位で受信がある．

NICT ダークネットデータの 7 月 11 日第 13 時間帯 (29,277 パケット) のうち，NMF エンジンによって攻撃だと判定されたデータ (12,252 パケット) を正例とした．7 月 11 日はマルウェア Morto の 3389 ポートへの攻撃が最初に認識された日である．また，同時間帯に正例だと判定されなかったデータを負例として用いる．ただし，計算時間の観点からそれぞれのデータをランダムサンプリングをして用いた．正例は 131 パケット，負例は 137 パケットである．ラベル無しデータは同日の第 14 番時間帯から第 48 番時間帯までの 6,314 パケットを用いた．実験に用いた正例，負例，ラベル無しデータの詳細を次表のとおりである．

| | 全パケット数 | ポート番号毎のパケット数 | | |
|---|---|---|---|---|
| | | 22 番 | 23 番 | 3389 番 |
| 無ラベル全体 | 6314 | 1032 | 2002 | 11 |
| 正例 | 130 | 37 | 62 | 0 |
| 負例 | 137 | 0 | 3 | 0 |

この実験の目的は，Morto による 3389 ポートへの攻撃を発見しやすくすることが目標であり，3389 ポートへのパケットを削除することなく，既知の 22, 23 番ポートへの攻撃によるパケットを削除することを目的としている．このとき，NMF エンジンは 22, 23 番ポートへの攻撃を検知し，警告を出していた．

**評価指標 1**: ポート番号 22, 23, 3389 番ポートへの攻撃パケットの残存率を精度の指標とする．スクリーニングの対象となるデータ $A$ のスクリーニング前のデータ数を $N$，スクリーニングの対象となるデータ $A$ のスクリーニング後のデータ数を $n$ とすると，データ $A$ の残存率を $n/N$ と定める．攻撃と認識されているパケットの残存率は少ないほうがよい．それらのパケットは，新しい攻撃を検知する際に必要ではないためである．また，攻撃とは関係ないと判断されたパケットの残存率は低いほうがよい．それは，新しい攻撃の検知に必要なデータを削除する可能性が低いことを意味するからである．

スクリーニングによる残存率の実験結果を表 2 に示す．実験に用いた各データの総数，そのうち新しく正ラベルをつけたデータ数を新正ラベルとし，スクリーニングによる残存率のデータを表にまとめる．

実験の結果から，3389 番ポートの残存率は，他の 2 つのポートの残存率よりきわめて高い．従って，当初の目的である Morto による 3389 ポートへの攻撃を発見しやすい環境にすることができていることがわかる．一方，無ラベル全体の残存率は低い．これはラベル有りデータとして用いた NMF エンジンの出力には，22, 23 番ポート以外のポートへの攻撃も含まれているためで，スクリーニングのポート番号毎の残存率が，ラベル有りデータに依存する傾向にあることを示している．

### 4.2 2014 年 1 月の国際連携データ

次に，世界各地のダークネットで観測されたトラフィックデータのうち，モルディブに設置されたダークネットの 2014 年 1 月 29 日のデータを用いた実験について報告する．本データでは，$SYN + ACK$ や $RST$ フラグ付きのパケットは攻撃とは関係ないため削除している．

**ラベル有りデータ固定実験**

正ラベル付きデータは，NM エンジンが攻撃であると判断した第 2 時間帯のパケット群から，25 %をランダムサンプリング (149 データ) した．また，負ラベル付きデータは，同時間帯の同パケット群以外のデータから 25 %をランダムサンプリング (170 データ) した．ラベル無しデータは，同日の第 3 時間帯から第 18 時間帯までを用いた．本実験では，次の指標で評価する．

**評価指標 2**: 次に NMF エンジンが攻撃だと判定したラベル有りデータとの一致度を評価指標とする．これには学習理論での評価指標である適合率，再現率，F 値を用いる．NMF エンジンが攻撃だと判定したデータの総数を $P$ 個，アルゴリズム MSSL が正ラベルだと判定したデータの総数を $p$ 個とする．また，どちらのデータにも含まれているパケットの総数を $a$ とする．このとき，適合率 $t = a/p$，再現率 $s = a/P$，F 値 $f = 2ts/(t + s)$ とする．

NMF エンジンが攻撃であると判断した 2014 年 1 月 29 日のデータは第 2 時間帯から第 18 時間帯までである．そのため，第 2 時間帯をラベル有りデータとして用い，他の時間帯のデータをラベル無しデータとして評価用とした．その結果，全体のラベル無しデータでの適合率，再現率，F 値は，それぞれ，83.40%，42.1%，57.1% となった．これらの評価値は必ずしも良いものとは言えない．原因として，ラベル無しデータ数が少ないため，十分に半教師あり学習の良さを活かせていないことがあげられる．また，ラベル有りデータの時間帯を固定していることも一因で，常時入れ替わる既知のマルウェアの攻撃の動きに学習が対応できないため，精度を悪くしていると考えられる．よってラベル有りデータを逐次更新して実験を行い結果を比較する．

**ラベル有りデータ逐次更新実験**

マルウェアによる攻撃は，比較的短時間で大量の亜種が発生する．実験から，学習に用いたデータから時間が経つと，半教師あり学習の精度が悪くなる．そこで，ラベル有りデータとして用いるデータを第 2 時間帯のみではなく，ラベル付けするデータの 30 分前（1 時間帯前）のデータを用いる．このことより新しい攻撃の特徴を捉えることで，精度の向上が期待される．本実験では，次の手順でアルゴリズム MSSL の実験評価を行った．$(i = 2, 3, \ldots, 17)$

(1) 第 $i$ 時間帯における NMF エンジンによるラベル有りデータを $L_i$ とする．

(2) 第 $i+1$ 時間帯のラベル無しデータ $U_{i+1}$ に対して，$L_i$ を用いてアルゴリズム MSSL を適用し，$U_i$ の新ラベ

| 実験結果 / 実験対象データ | 総数 | 新正ラベル | スクリーニング後の残存率 |
|---|---|---|---|
| 無ラベル全体 | 355495 | 222327 | 37.46 % |
| 22 番ポート | 35242 | 29013 | 17.64 % |
| 23 番ポート | 8657 | 8513 | 1.66 % |
| 3389 番ポート | 16893 | 777 | 95.40 % |

**表 2** アルゴリズム MSSL によるスクリーニング実験結果 (残存率)

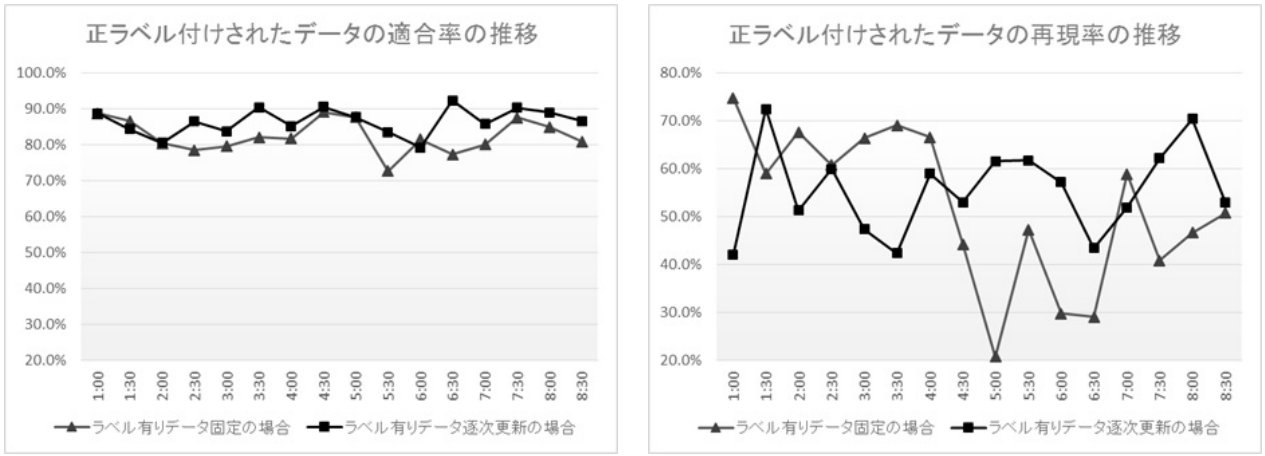


**図 1** アルゴリズム MSSL のラベル有りデータ固定時と逐次更新時の適合率 (左図) と再現率 (右図) の推移．横軸はデータを 30 分毎に分割した際の開始時間を示している．

ル有りデータ $U_i'$ を得る．

(3) 第 $i+1$ 時間帯における NMF エンジンによるラベル有りデータ $L_{i+1}$ と $U_i'$ を評価指標 2 を用いて評価する．

30 分毎にラベル有りデータを更新したときのラベル無しデータ全体の適合率は 86.77% となった．これは，正ラベル付きデータを第 2 時間帯に固定したときの適合率 83.40 よりも良い．また，再現率や F 値も平均して向上している．適合率と再現率のラベル有りデータ固定時と逐次更新時の推移を図 1 にあげる．このことから常にラベル有りデータを更新する方が効果的に学習できると結論付けられる．

## 5. まとめ

本論文では，グラフベースの半教師あり学習を用いたスクリーニング手法を提案した．さらにその有効性を，NICT 及び国際連携によるダークネット観測網と NMF エンジンによって得られたラベル有り/ラベル無しデータ上で実証した．本手法の欠点の一つは半教師あり学習によるラベル付けの再現率が低いことである．これを解決するために，頻出系列パタンスクリーニングとの連携を行うことが今後の課題である．

**謝辞** 本研究の一部は総務省による「国際連携によるサイバー攻撃の予知技術の研究開発」の支援を受けたものである．また，使用したデータは，独立行政法人・情報通信研究機構 (NICT) からダークネット観測データの提供を受けたものである．また，本研究の一部は JSPS 科研費 26280087 の助成を受けたものである．

## 参考文献

[1] Avrim Blum and Shuchi Chawla. Learning From Labeled and Unlabeled Data Using Graph Mincuts. *Proc. ICML*, pp.19–26, 2001.

[2] Masashi Eto, Daisuke Inoue, Mio Suzuki, and Koji Nakao. A Statistical Packet Inspection for Extraction of Spoofed IP Packets on Darknet.*Proc. 4th Joint Workshop on Information (JWIS 2009)*, 2009.

[3] 川村 勇気, 島村 隼平, 中里 純二, 吉岡 克成, 衛藤 将史, 井上 大介, 竹内 純一, 中尾 康二. 非負値行列分解を用いたボットネット検出実験. 信学技報 113(288), ICSS2013-61, pp.23–28, 2013.

[4] 川村 勇気, 川喜田 雅則, 村田 昇, 竹内 純一. 非負値行列因子分解における MDL 原理について. 第 37 回情報理論とその応用シンポジウム予稿集 (SITA 2014)，pp.518–523, 2014.

[5] Atsushi Okamoto and Takayoshi Shoudai. Mining First-Come-First-Served Frequent Time Sequence Patterns in Streaming Data. Proc. IADIS International Conference on e-Society (ES2013), pp.283–290, 2013.

[6] Amar Subramanya and Partha Pratim Talukdar. Graph-Based Semi-Supervised Learning, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2014.

[7] Hisashi Tsuruta, Takayoshi Shoudai, and Jun'ichi Takeuchi. Network Traffic Screening Using Frequent Sequential Patterns. Intelligent Control and Innovative Computing, Springer, Lecture Notes in Electrical Engineering, Vol.110, pp.363–375, 2012.

[8] Xiaojin Zhu and Andrew B. Goldberg. Introduction to Semi-Supervised Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2009.