2016年度 修士論文

ハニーポットを用いた DRDoS攻撃検知法

提出日：2017年1月30日

指導：後藤滋樹教授

早稲田大学　基幹理工学研究科　情報理工・情報通信専攻
学籍番号：5115F037-1

篠宮 一真

目次

図一覧

表一覧

第 1 章

序論

1.1 研究の背景

情報通信技術の発達に伴い, 企業, 政府機関, 教育・研究機関等の様々なサービスが電子化されインターネットを介して提供されるようになった. しかしこれらの動きに伴い, ホストやネットワークに過負荷を与えてサービスの提供を妨げる DoS 攻撃[1]が問題となっている. その中でも特に, インターネット上の様々なサービスを踏み台として悪用する DRDoS [2] 攻撃の脅威が拡大している.

DRDoS 攻撃とは, インターネット上に公開されているサーバを踏み台にして大量のパケットを攻撃対象組織に送信することにより, その組織のネットワーク等のリソースを圧迫する攻撃である. DRDoS 攻撃では, 複数の踏み台を利用することにより攻撃者が攻撃の通信量を非常に大きくすることが可能であり, 2014 年に実行された攻撃では最大で約 400Gbps, 2015 年には約 500Gbps もの攻撃通信が観測されている [1]. また, DDoS 攻撃の件数も年々増加している [2].

このように, DRDoS 攻撃による被害は年々その深刻さを増しており, さらに Booter や Stresser と呼ばれる DDoS 攻撃代行サービスも登場し, 攻撃に関する知識を持たない者でも DRDoS 攻撃を容易に実行できるようになった. また, 国際的なハッカー集団や, 企業を脅迫して身代金を要求する攻撃活動においても, DRDoS 攻撃が攻撃の実行手段として利用されている. これらの事例から, 今後も DRDoS 攻撃による脅威は拡大することが予想される.

しかし, DRDoS 攻撃の被害を完全に防ぐ方法は確立されておらず, 一度攻撃が実行されると被害者側はその攻撃が終わるのを待つか, ブラックホールルーティングやパケットフィルタリング等の技術を利用して被害を軽減させながら耐えるしかないのが現状である. このような状況の中で ISP (Internet Service Provider) が通信事業用設備を維持しサービスを安定的に提供するためには, 早期に障害発生等の原因となる攻撃を把握し対応することが攻撃の被害を軽減

[1]Denial of Service の略であり, サービス拒否攻撃を意味する
[2]Distributed Reflection DoS の略であり, 分散反射型サービス拒否を意味する

させるための重要な要素である.

ネットワークを運用する ISP では, DoS 攻撃の脅威からインフラ設備をの被害を緩和するために, バックボーンネットワークに配置した DoS 攻撃対策システムによる常時監視や大量通信に対する規制等を実施している. その際, ISP バックボーンのトラヒックを監視には大量のデータを効率よく監視するために, フローデータを収集して攻撃検知を行う. フローデータには本来のデータの一部の情報しか含まれていないため, 定常的な通信監視による DRDoS 攻撃は, 検知をしようとしても攻撃であるか否かを判断するのが困難である. また攻撃検知の条件としてトラフィック量の閾値超過が利用され, その閾値は誤検知低減のため高く設定されるるため, 閾値に到達するまで検知が遅延するという問題点がある.

1.2 研究の目的

以上の背景を踏まえ, 規模が拡大する DRDoS 攻撃に対してネットワーク管理者がより早期な対応ができるよう, パケットレベルで機械学習を活用して DRDoS 攻撃を検知することにより高精度かつ早期の攻撃検知法を提案する.

本研究では DRDoS ハニーポットと呼ばれる, DRDoS 攻撃に利用される複数種類のネットワークサービスを提供するサーバを日米の 3ヶ所に囮 (おとり) としてインターネット上に設置した.

ハニーポットは踏み台として動作し, 攻撃をパケットレベルで観測可能であるため, ISP バックボーンで収集しているフロー情報よりも詳細な攻撃特徴量が抽出できる. それを活かすことで高精度かつ早期の攻撃検知を目指す.

1.3 本論文の構成

本論文は以下の章により構成される.

第 1 章 序論

本研究の背景と概要を述べる.

第 2 章 DRDoS 攻撃とその対策法

DRDoS 攻撃と防御法について解説する.

第 3 章 関連研究と理論

本研究の関連研究および, 本研究において使用する機械学習の理論を説明する.

第 4 章 ハニーポットによるデータ収集

ハニーポットによるデータ収集の方法と収集データの統計情報を示す.

第 5 章 提案手法

本研究の提案手法を詳細に説明する.

第 6 章 実験結果

実験結果を示し, 考察する.

第 7 章 まとめと今後の課題

本研究をまとめ, 残された今後の課題について述べる.

第 2 章

DRDoS 攻撃とその対策法

本章では, DRDoS 攻撃について詳しく説明した後, その対策法と現状の問題点について述べる.

2.1 DRDoS 攻撃

DRDoS 攻撃とはインターネット上に存在する複数のリフレクタと呼ばれるサーバを踏み台としたリフレクション攻撃を一斉に同一の攻撃対象に仕掛けるものである. 多量のパケットを攻撃対象に送信することにより, 通信帯域等のリソースを圧迫する. 図 2.1 にリフレクション攻撃の概要を示す.

DRDoS 攻撃では, リフレクタにおける次の 2 つの性質が悪用される [3].

1 つ目は増幅効果である. これは要求のパケットサイズよりも応答のパケットサイズが大きくなる性質であり, これ悪用することにより, パケットの送信者は小さなサイズのパケットから大きなサイズのパケットを発生させることが可能である.

2 つ目は反射効果である. これは通信相手との接続を確認せずにコネクションレスな通信を行なう性質である. 要求パケットの送信元 IP アドレスを確認しないプロトコルを使用することにより, 攻撃者は応答パケットを任意のホストへ送信させることができるため, パケットの送信者は送信元 IP アドレスを容易に詐称することが可能である.

DRDoS 攻撃は, 複数のリフレクタに対して攻撃者が送信元 IP を攻撃対象者のものに偽装したパケット送信し, そのレスポンスで対象者に大量トラフィックを送信する. リフレクタには, コネクションを必要としないプロトコルである UDP のサービスが悪用されることが多く, DNS, NTP, SSDP などの 14 種類のプロトコルが DRDoS 攻撃に悪用された場合に, 高い増幅効果をもつ [5]. 以下, 本研究で対象としている 5 種類のプロトコルを悪用した攻撃について説明する.

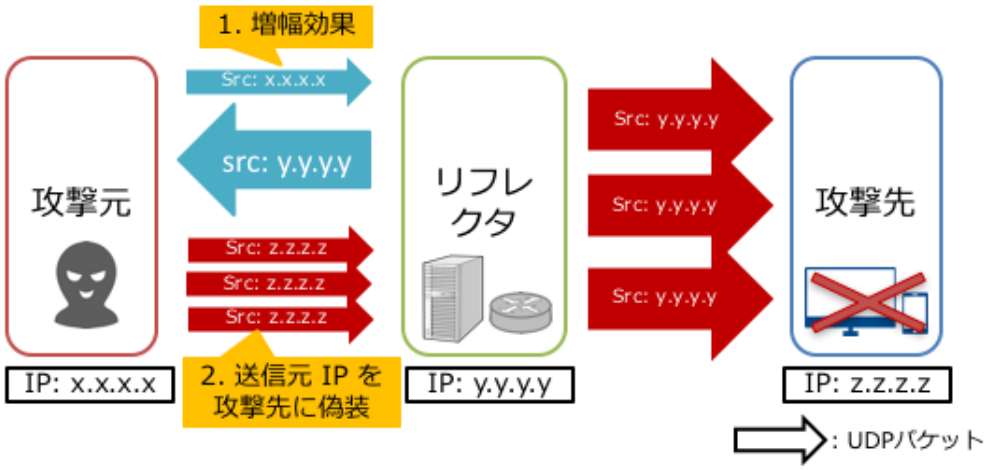

図 2.1: リフレクション攻撃

2.1.1 NTP リフレクション攻撃

NTP (Network Time Protocol) は UDP123 番ポートを使用し, システムの時刻を同期させるためのプロトコルである. NTP サーバは一般的に公開されているため容易に発見することが可能であり, NTP サーバの状態を確認するための機能である monlist コマンドを用いてパケットを増幅させることができる. monlist コマンドはサーバが過去に通信を行った端末の最大 600 台分の履歴を要求する機能であり, これを悪用することで 556.9 倍の増幅率が得られる [4].

2.1.2 RIP リフレクション攻撃

RIP (Routing Information Protocol) は UDP520 番ポートを使用するルーティングプロトコルの一つであり, ルータ等の通信機器間での経路情報のやりとりや経路の決定を行う際に利用される. ルータは起動時にブロードキャストで周りの機器にルーティング情報を要求し, 要求を受け取った機器はすべて応答を返す. この時の要求に対する応答が大きいことが悪用され, 増幅率は 131.2 倍である [4].

2.1.3 chargen リフレクション攻撃

chargen (character generation protocol) は UDP19 番ポートを使用するプロトコルであり, 主にネットワークの動作確認や性能測定, デバッグなどで用いられる. 通信が確立すると文字列データを自動生成し, 切断されるまで文字列データを送り続ける. クエリパケット内のデータは破棄されるため, クエリパケットのペイロード長を短く設定することで増幅率が大きくなり, その増幅率は 358.8 倍である [4]. chargen は運用中のサーバやネットワークではあまり使われないため, サービスの停止を促すことが対応策として求められる.

2.1.4 SNMP リフレクション攻撃

SNMP (Simple Network Management Protocol) は UDP161 番ポートを使用するプロトコルであり, ネットワーク経由で機器を監視・制御するためのプロトコルである. ネットワーク管理者が利用する, 機器の管理・監視のためのコンピュータやソフトウェアを SNMP マネージャ, 監視・制御下に置かれる機器やソフトウェアを SNMP エージェントと呼び, 両者が SNMP で通信を行い監視・制御を行う. SNMP マネージャはネットワーク上の複数の管理情報をまとめて要求する GetBulkRequest を利用することで, 増幅率が約 6.3 倍の応答が得られる [4] ことが悪用される. 特徴として, コミュニティ名にデフォルト値である “public” が使用されていることがあり, 対策としてこのコミュニティ名を変更することが挙げられる.

2.1.5 SSDP リフレクション攻撃

SSDP は UDP1900 番ポートを使用するプロトコルであり, UPnP (Universal Plug and Play) による通信に利用される. UPnP は LAN 内に存在する UPnP 対応機器の発見や操作を行うものであり, SSDP は機器同士がお互いを発見させるプロセスで用いられる. 攻撃者は M-Search と呼ばれる SOAP (Simple Object Access Protocol) リクエストを UPnP 対応機器に向けて応答パケットの増幅率が上がるように送信する. これにより 30.8 倍の増幅率が得られる [4].

2.2 DRDoS 攻撃の対策法と問題点

ネットワークを運用する ISP では, DoS 攻撃の脅威からインフラ設備をの被害を緩和するために, バックボーンネットワークに配置した DoS 攻撃対策システムによる常時監視やその原因となる大量通信に対する規制等を実施している.

DDoS 攻撃の複雑化に伴い, DPI (Deep Packet Inspection) により攻撃トラヒックを識別するシステムなど, 高機能化が進んでいるが, ISP がバックボーン上の全トラヒックを解析するにはコストがかかる. そのため一般的に, ISP バックボーンのトラヒックを監視する場合, 大量のデータを効率よく監視するためにフローデータを収集して攻撃を検知する. フローデータを利用し, ネットワーク設備に影響を与える可能性のある大量通信を検知し, その後検知したトラヒックに対して, DPI による詳細解析やブラックホール・ルーティング等を行うことで効率的に対処している.

しかし, フローデータには本来のデータの一部の情報しか含まれず, 詳細なトラヒック分析が困難である. また, パケットの観測後にフローとして集約し送信する時間や, 複数のフローを集約する時間により検知の遅延が発生する. また正常な通信も流れているため, 定常的な通信監視による DRDoS 攻撃の検知は, 攻撃の判断が困難でその処理に時間を要するという問題点

がある. また, 攻撃検知の条件として, トラフィック量の閾値超過が利用されるが, その閾値は誤検知を低減するために高く設定する必要があり, 閾値に到達するまで検知が遅延するという問題点がある.

第 3 章

関連研究と理論

本章では, 関連研究について述べた後, ハニーポットを用いて収集したデータを分析するための機械学習手法について説明する.

3.1 関連研究

牧田 [3, 5] は DRDoS ハニーポットを構築し, マルウェアによる DRDoS 攻撃やリフレクタの視点での DRDoS 攻撃の観測, それらの分析を統合して行うシステムを運用している. 西添 [6] は DRDoS 攻撃に利用される複数種類のネットワークサービスを提供するサーバを囮としてインターネット上に設置し, その通信を観測することで DRDoS 攻撃を観測する手法を提案し, 攻撃の分析を行っている.

また, 牧田 [7] はハニーポットを用いて DRDoS 攻撃アラートシステムを構築している. 受信時刻の間隔が閾値以下のグループごとに, 流量が閾値を超えるものを DRDoS と判定してアラートを送信している. 蒲谷 [8] は同一 IP アドレスから 60 秒以上の間 隔をあけずに 100 パケット以上のパケットを受信 した場合, その一連の通信を該当 IP アドレスに対する攻撃と見なし, アラート情報を送信している.

浦川 [9] は DNS アンプ攻撃を対象とし, DNS ハニー ポットで収集した通信と ISP バックボーンにおけるフローデータの突合分析を行い, ハニーポット監視による攻撃の早期検知および規模推定の実現性について検証した. ハニーポットを監視することで, 攻撃事例の約 75%を既存の DoS 攻撃対策システムよりも早期に検知可能であることを示している.

柴原 [10] は DRDoS 攻撃を観測可能なダークネットを用いて, 攻撃に悪用される DNS, NTP, SSDP のリフレクタの分析を行った. また, 牧田 [11] は DNS ハニーポットが観測した DNS アンプ 攻撃とダークネットで観測した DNS サーバのスキャン活動の相関を分析し, DNS アンプ攻撃が DNS ハニーポットで観測される前に, 攻撃と同じドメイン名を用いたスキャン活動がダークネットセンサでも観測される可能性が高いことを確認した.

Santanna[12] は Booter と呼ばれる DDoS 攻撃の代行サービスを自分らのインフラに向けて利用することでその特徴を分析し, 対策法について論じている. Karami[13] も DDoS 攻撃の代行サービスについて, その対価の支払いの仕組みや, 実際に使われている攻撃ツールの分析を行っている.

3.2 機械学習

パターン認識の学習, 言語の文法の学習, ロボットの行動の学習など, 様々な種類の学習課題を対象として, 学習アルゴリズムの研究をする分野を機械学習という. 機械学習には教師なし学習と教師あり学習がある. 教師あり学習とは, 学習データ (入力と出力のペア) が有限個与えられたとき, その情報に基づいて新しい入力に対して正しい出力を予測することである. パターン認識や回帰分析がこれにあてはまる. 教師なし学習は, 出力がない訓練データから何らかの有用な情報を導き出すことが目的であり, クラスタリングや主成分分析などがあてはまる. 後述する決定木や SVM は教師あり学習, SOM は教師なし学習の一手法である.

本節では, セキュリティ分野の研究においてによく用いられている機械学習である決定木, SVM, SOM の概要を述べる.

3.2.1 決定木

決定木 (Decision Tree) とは図 3.1 のような木構造をした, 決定や分類を行うためのグラフであり, 非線形判別分析の一つである. 説明変数の値のある基準を基に分岐させ, 判別のモデルを構築する. 分岐の過程は木構造で図示可能であり, if 文のような簡潔な規則で記述可能である. 判別が高速であり, 人間が理解しやすいことも決定木の特徴の一つである.

与えられたデータから適切な決定木を作成する事を決定木の構築と呼ぶ. 決定木構築アルゴリズムは代表的なものに CHAID, C4.5/C5.0/See5, CART がある.

CHAID (CHi-squared Automatic Interaction Detection) は Morgan が提案した AID (Automatic Interaction Detection) を 1975 年に Hartigan が発展させたものであり, カイ 2 乗統計量や F 統計量が分岐基準として用いられている.

C4.5/C5.0/See5 はオーストラリアの J. Ross Quinlan が 1986 年に提案した ID3 (Interactive Dichotomiser 3) を発展させたものである. ID3 は分岐基準として情報利得 (information gain) を用いているのに対し, C4.5/C5.0/See5 は利得比を用いていおり, 2 進木に限らないという特徴を持つ.

CART は説明変数を 2 進木に分岐させ, 分岐基準として経済学者ジニが提案したジニ係数を使うジニ多様性指標 (Gini's diversity index) や利得比 (gain ratio) が用いられている. CART では決定木を予め無制限に生長させ, ある基準に基づいて枝刈りを行うことで決定木を構築する.

これらのアルゴリズムの大きな違いは決定木の生成・生長, 枝刈りのアルゴリズムである. 決定木の生成・生長とは, データセットから決定木の幹や枝となる説明変数を選定し, 分岐基準を基に分岐させ, 木を生長させることである. 木を生成する際にどの変数のどの値を木の分岐点にするかに関して計算方法が異なる.

鈴木 [14] が行った各決定木構築アルゴリズムの比較によると, 精度の点では CART が最も良いが計算量や使用するメモリ量が大きいため, 本研究に CART は適していないと考えられる. C4.5 は ID3 は不可能な連続値の取扱いが可能であるため本研究に適用可能であるため, 本研究では決定木構築アルゴリズムとして C4.5 を採用する.

なお, 本研究では C4.5 の実装としてデータマイニングツールの Weka [15] に含まれる J48 を用いる. Weka とはニュージーランドの Waikato 大学によって, Java を用いて開発されたデータマイニングツールであり, 日本国内での研究における適用例も多い.

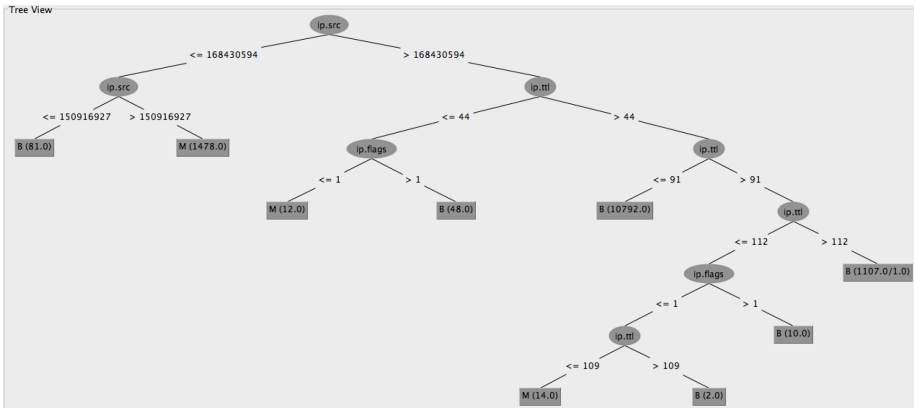

図 3.1: J48 による決定木の例

3.2.2 サポートベクターマシン

サポートベクターマシン (SVM: Support Vector Machine) はニューロンの最も単純な線形閾素子を拡張した, 優れた識別機能を有する教師あり学習の一つである. 初期値によって最適解が異なるという局所解の問題が無く, 局所的最適解が必ず大局的最適解になるという利点を持つ. SVM は線形識別器であるが, カーネル関数を用いることで非線形識別器に拡張できる. これにより, 複雑な境界面の識別が可能である. 以下, SVM の理論について説明する [16, 17, 18, 19].

ハードマージン SVM

d 個の特徴量による判別データ $x = (x_1, \ldots, x_d)^T$ の識別関数は, w_i を線形 SVM の重みパラメータ, $\boldsymbol{w}$を重みベクトル, b をバイアス項とするとき, 次のように表される.

$$f(\boldsymbol{x}) = \sum_{j=1}^{d} w_j x_j + b \tag{3.1}$$

x は d 次元空間における点であり, f の正負によって正常/不正の 2 つのクラスのいずれかに分類される. この線形 SVM の $f(x) = 0$ を満たす点の集合は $d-1$ 次元の超平面となり, 2 つのクラスの境界面を成す. この境界面は重みベクトル $\boldsymbol{w}$ を変えることによって制御できる. 図 3.2 のように 2 つのクラスの学習データが $d-1$ 次元の超平面で完全に分離可能な場合を線形分離可能といい, このような SVM をハードマージン SVM という.

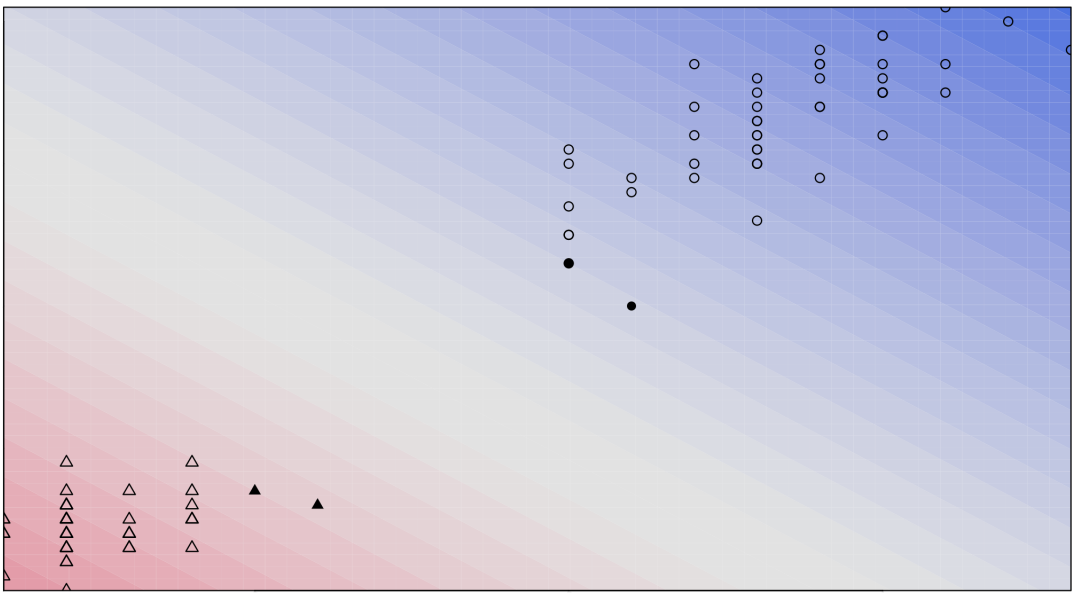

図 3.2: ハードマージン SVM による識別の例

学習サンプルを $x_1, \ldots, x_l$, それぞれのクラスを $y_1, \ldots, y_l$ と表す. 例えば, 学習データ x_1 が正常である場合, $y_1 = 1$ となる. パラメータ$\boldsymbol{w}$, b は定数倍しても超平面が変わらないという冗長性持っているため, 次式の制約によって学習結果を一意に定める.

$$\min_{i=1,\ldots,l} |\boldsymbol{w}^T \boldsymbol{x}_i + b| = 1 \tag{3.2}$$

学習サンプルと超平面の距離はヘッセの公式より $\frac{|\boldsymbol{w}^T \boldsymbol{x}_i + b|}{\|\boldsymbol{w}\|}$ で与えられる. 学習データを完全に識別できる超平面は無数に存在するが, 最も良い識別面となる超平面を求めるためには, 超平面と学習データの最小距離を最大化すればよい. これは, 次式を満たす$\boldsymbol{w}$, b を求めることになる.

$$\max_{\boldsymbol{w},b}\left[\min_{i=1,\ldots,l}\frac{|\boldsymbol{w}^T\boldsymbol{x}_i+b|}{\|\boldsymbol{w}\|}\right] \tag{3.3}$$

ここで式 (3.2) の制約によって $\frac{1}{\|\boldsymbol{w}\|}$ となるため, $\boldsymbol{w}$, b は式 (3.4) によって学習データを完全に識別するものの中から最小距離を最大化するように決定する. なお, 制約条件は超平面が学習データを完全に識別できることを表す.

$$\begin{cases} \text{目的関数:} & \min_{\boldsymbol{w}}\|\boldsymbol{w}\|^2 \\ \text{制約条件:} & y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\geq 1 \quad (i=1,\ldots,l) \end{cases} \tag{3.4}$$

このように, SVM の学習は線形制約条件付きの二次計画問題に帰着する. この式を主問題という. これを双対問題に変換するためにラグランジュ乗数 α ($\alpha_i \geq 0$) を導入することで, 次式が得られる.

$$L(\boldsymbol{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{l}\alpha_i\{y_i(\boldsymbol{x}_i^T\boldsymbol{w}+b)-1\} \tag{3.5}$$

この最適化問題を解くためには, α を最大化, L を $\boldsymbol{w}$,b に関して最小化すればよい. 最適解においては L の勾配が 0 になるため, 以下の式が得られる.

$$\frac{\partial L(\boldsymbol{w},b,\boldsymbol{\alpha})}{\partial \boldsymbol{w}} = \sum_{i=1}^{l}\alpha_i y_i\boldsymbol{x} + \boldsymbol{w} = 0 \tag{3.6}$$

$$\frac{\partial L(\boldsymbol{w},b,\boldsymbol{\alpha})}{\partial b} = -\sum_{i=1}^{l}\alpha_i y_i = 0 \tag{3.7}$$

これらより, 次式が成り立つ.

$$\boldsymbol{w} = \sum_{i=1}^{l}\alpha_i y_i \boldsymbol{x}_i \tag{3.8}$$

$$\sum_{i=1}^{l}\alpha_i y_i = 0 \tag{3.9}$$

式 (3.8), (3.9) を式 (2.7) に代入することで, 式 (3.4) の双対形式が得られる.

$$\begin{cases} \text{目的関数:} & \max_{\boldsymbol{\alpha}}\sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^T\boldsymbol{x}_j \\ \text{制約条件:} & \sum_{i=1}^{l}\alpha_i y_i = 0, \quad \alpha_i \geq 0 \quad (i=1,\ldots,l) \end{cases} \tag{3.10}$$

これは $\boldsymbol{\alpha}$ のみに関する最大化問題であり, 解 $\hat{\alpha}_i$ が求まれば, 式 (3.8) より $\hat{\boldsymbol{w}}$ が求まる. この凸 2 次計画問題により, 局所的最適解が大局的最適化になり, 局所的最適解の問題を避けることができる. 主問題の式 (3.4) では $\boldsymbol{x}_i$ が単独で用いられているが, 双対問題の式 (3.10) に変換することにより, $\boldsymbol{x}_i^T\boldsymbol{x}_j$ のような内積の形になる. これによって後に述べるカーネル関数が定義できる. 主問題の最適解 $\hat{\boldsymbol{w}}, \hat{b}$ と双対問題の最適解 $\hat{\boldsymbol{\alpha}}$ は以下の KKT (Karush- Kuhn-Tucker) の相補条件を満たさなければならない.

$$\hat{\alpha}_i\left[y_i\left(\hat{\boldsymbol{w}}^T\boldsymbol{x}_i+\hat{b}\right)-1\right]=0 \tag{3.11}$$

図 3.2 において, 超平面 $\boldsymbol{w}^T\boldsymbol{x}_i+b=0$ から最短距離にある, 黒く塗りつぶされているサンプルのみが目的関数を最大化している. これらはの $\boldsymbol{x}_i$ は $\alpha_i>0$, すなわち, $y_i(\hat{\boldsymbol{w}}^T\boldsymbol{x}_i+\hat{b})-1=0$ となって超平面上にあり, 識別関数を支持しているため, サポートベクターという. 多くの α_i は $\alpha_i=0$ となるため, サポートベクターとなるサンプルは少なく, 計算量の節約になる. サポートベクターのみが境界の決定に影響することをスパースネス (sparseness) といい, これは SVM の特長である. x_{+1}, x_{-1} をそれぞれ異なるクラスのサポートベクターとすると, バイアス項 b は, 主問題の制約式を用いて次式によって得られる.

$$b=-\frac{1}{2}\left(\boldsymbol{w}^T\boldsymbol{x}_{+1}+\boldsymbol{w}^T\boldsymbol{x}_{-1}\right) \tag{3.12}$$

ソフトマージン SVM

ここまでは学習データが超平面によって完全に分離できるハードマージンを仮定してきた. しかし現実問題ではクラスの重なりがあり, 超平面によって完全には分離できないため誤判定を許す. 完全分離できない場合は式 (3.4) の制約条件を満たす $\boldsymbol{w}$,b が存在せず最適化ができない. このような場合, マージンのほかのクラスの側に他クラスのデータ点があってもよい. これをソフトマージンといい, この場合の SVM をソフトマージン SVM という. 図 3.3 にソフトマージン SVM による識別の例を示す.

ソフトマージンでは最適化を行うための制約条件を緩めるために, スラック変数 $\xi_i\geq 0$ $(i=1,\ldots,l)$ を導入し, 目的関数と制約条件を以下のように変更する. なお, このスラック変数 ξ_i は誤った領域に入る, すなわち誤判定される割合を意味する.

$$\begin{cases} \text{目的関数:} & \min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{l}\xi_i \\ \text{制約条件:} & y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\geq 1-\xi_i,\ \ \xi_i\geq 0\ \ (i=1,\ldots,l) \end{cases} \tag{3.13}$$

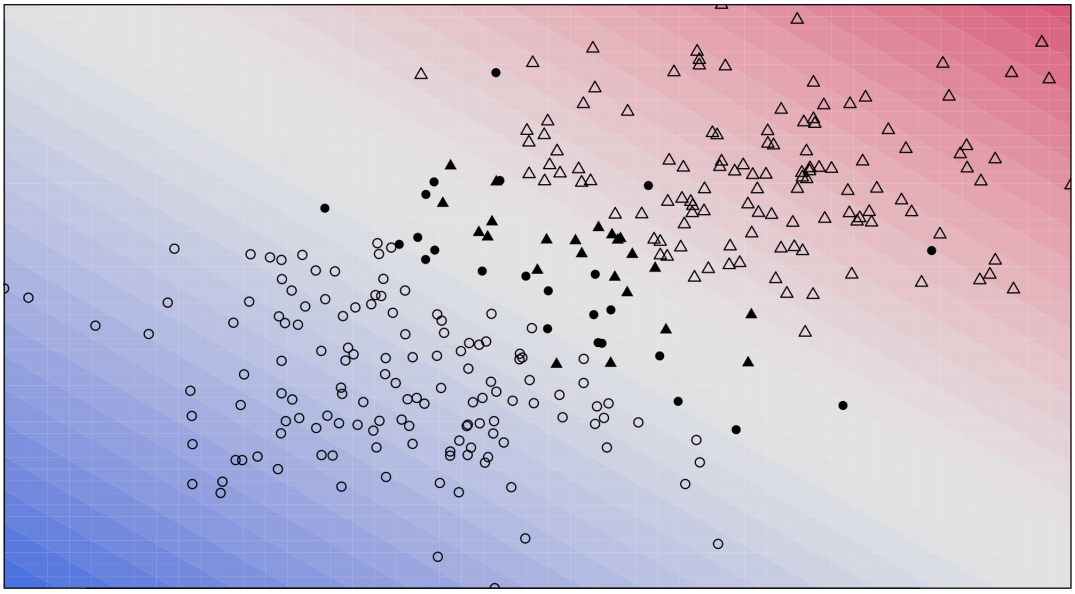

図 3.3: ソフトマージン SVM による識別の例

C は制約条件をどこまで緩めるかを制御するパラメータであり, 予め決めておく必要がある. これは第 4 章において述べる. このようにして最適化問題を変更すると, ラグランジュ乗数 α に関する問題は次式のようになる.

$$\begin{cases} \text{目的関数:} & \max_{\boldsymbol{\alpha}} \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j \\ \text{制約条件:} & \sum_{i=1}^{l} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C \quad (i = 1, \ldots, l) \end{cases} \tag{3.14}$$

カーネル法

ここまでは, 線形の識別問題に対して, ソフトマージンによってある程度の誤判定を許容することで線形分離可能としてきた. 線形識別器は 2 つのクラスが超平面で分離することができる場合は良い精度が得られるが, 常にそのようになるわけではない.

そこで, より高次元の特徴空間への写像 $\Phi : R^d \to R^D (d \ll D)$ を事前に行うことで線形分離性を高め, 写像先の空間 R^D において線形識別を行うカーネルトリックという方法を用いる.

カーネルトリックによる非線形 SVM による識別の例を図 3.4 に示す. これにより, 線形識別よりも複雑な識別面が表現可能となっていることがわかる.

この写像においては, 元の空間におけるデータ同士の距離関係をある程度保存する必要があるため, 元の空間で定義されるカーネル関数 $K(x, x')$ を用意する. このとき, Φ は次の条件を満たす.

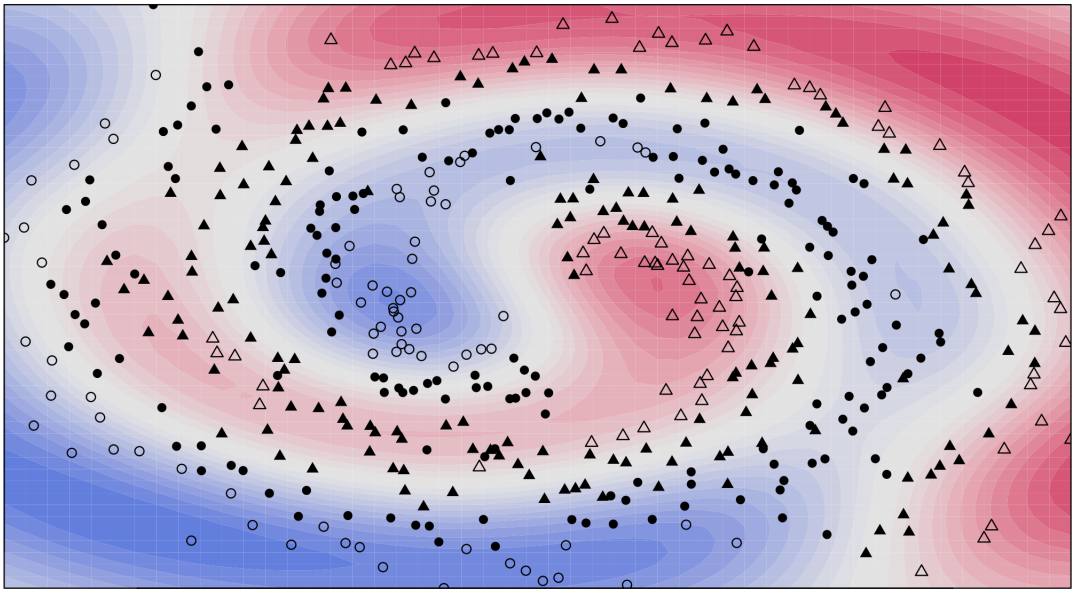

図 3.4: 非線形 SVM による識別の例

$$K(\boldsymbol{x}, \boldsymbol{x'}) = \Phi(\boldsymbol{x})^T \Phi(\boldsymbol{x'}) \tag{3.15}$$

K を 2 点間の近さを表す関数とすれば, 内積で近さが保存される. このような Φ が存在するとし, R^D において SVM を適用し, 式 (2.12) を用いると, 識別関数は次のようになる.

$$\begin{aligned} f\Big(\Phi(\boldsymbol{x})\Big) &= \boldsymbol{w}^T \Phi(\boldsymbol{x}) + b & (3.16)\\ &= \sum_{i=1}^{n} \alpha_i y_i \Phi(\boldsymbol{x})^T \Phi(\boldsymbol{x}_i) + b & (3.17)\\ &= \sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b & (3.18) \end{aligned}$$

また, 学習の問題も同様にして, 以下のように Φ を用いずに K のみで記述できる.

$$\begin{cases} \text{目的関数:} \quad \max_{\boldsymbol{\alpha}} \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{制約条件:} \quad 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{l} \alpha_i y_i = 0 \quad (i = 1, \ldots, l) \end{cases} \tag{3.19}$$

したがって, R^D における識別が Φ の求解をせずに行える. この方法をカーネルトリックという. カーネル関数 $K(\boldsymbol{x}, \boldsymbol{x'})$ には, d 次多項式カーネル, シグモイドカーネルなどがあるが, 本研究では次式で表されるガウシアンカーネルを用いる. γ はガウシアンカーネルを用いた場合のパラメータとして設定する必要がある.

$$K\left(\boldsymbol{x}, \boldsymbol{x'}\right) = \exp\left(-\gamma\|\boldsymbol{x} - \boldsymbol{x'}\|^2\right) \quad \left(\gamma = \frac{1}{\sigma^2}\right) \tag{3.20}$$

SVM の実装

オープンソースの機械学習ライブラリとして配布されている SVM の実装の代表的なものには LIBSVM [20], SVM light [21], R [22] などがある．本研究では SVM の実装として利用実績の多い LIBSVM を用いる．LIBSVM は国立台湾大学で開発され, C 言語 API を用いた C++で記述されている．カーネル法を用いた学習に使う SMO アルゴリズムを実装しており, 統計分類と回帰分析に対応している．詳細については Chih-Chung Chang [20, 23] を参照されたい．

3.2.3 自己組織化マップ

自己組織化マップ (SOM: Self-Organizing Map) はフィンランドの科学者 Teuvo Kohonen によって提案された教師なし学習のニューラルネットワークアルゴリズムである．ニューラルネットワークの中ではフィードフォワード型に分類され, フィードフォワードニューラルネットワークとも呼ばれる．入力層と出力層 (競合層) の 2 層によって構成されるニューラルネットワークである．以下, SOM の理論について説明する [24]．なお, 本研究では SOM の実装に R の kohonen ライブラリ [25] を用いる．

入力層に分析対象の個体 j $(j = 1, 2, \ldots, n)$ の変数ベクトル $\boldsymbol{x}_j$ $(x_{j1}, x_{j2}, \ldots, x_{jp})$, 出力層には k $(i = 1, 2, \ldots, k)$ 個のユニット $\boldsymbol{m}_i$ があるとする．図 3.5 (a) に示すように, 出力層の任意のユニットは入力層の変数ベクトルのすべてとリンクしている．以下, SOM のアルゴリズムについて述べる．初期段階では図 3.5 (b) に示すように重み $\boldsymbol{m}_i$ $(m_{i1}, m_{i2}, \ldots, m_{in})$ が付けられている．

SOM のアルゴリズム

1. 式 (2.1) を用いて, 入力 x_j と出力層のすべてのユニットを比較し, 最も類似しているユニット m_c をそのユニットの勝者 (winner) とする

$$\|\boldsymbol{x}_j - \boldsymbol{m}_c\| = \min_i \|\boldsymbol{x}_j - \boldsymbol{m}_i\| \tag{3.21}$$

2. 勝者ユニット m_c およびその近傍のユニットの重みベクトル m_i を次式によって更新する

$$\boldsymbol{m}_i(t+1) = \begin{cases} \boldsymbol{m}_i(t+1) + h_{ci}(t)[\boldsymbol{x}_j(t) - \boldsymbol{m}_i(t)] & (i \in N_c) \\ \boldsymbol{m}_i(t) & (i \notin N_c) \end{cases} \tag{3.22}$$

$$h_{ci}(t) = \alpha(t) \exp\left(- \frac{\|r_c - r_i\|^2}{2\sigma^2(t)} \right) \tag{3.23}$$

式 (2.3) の $h_{ci}(t)$ は近傍関数であり, ユニット c とその近傍のユニット i の近さによって x_j への影響を調整する. $\alpha(t)$ は学習率の係数であり, r_c と r_i はユニット c と i の 2 次元上座標位置ベクトルである. $\alpha(t)$ と $\sigma^2(t)$ は学習回数を変数とする単調減少関数 $1 - \frac{t}{T}$ (t: 学習回数, T: 学習の総回数) である.

3. すべての入力の特徴ベクトル $\boldsymbol{x}_j$ $(j = 1, 2, \ldots, n)$ に対し, 1, 2 を繰り返す.

SOM は上記のアルゴリズムによって多次元空間上の分類対象を 2 次元空間に射影する. SOM の結果出力のユニットは多くの場合, 蜂の巣状に六角形のユニットを並べて構成される.

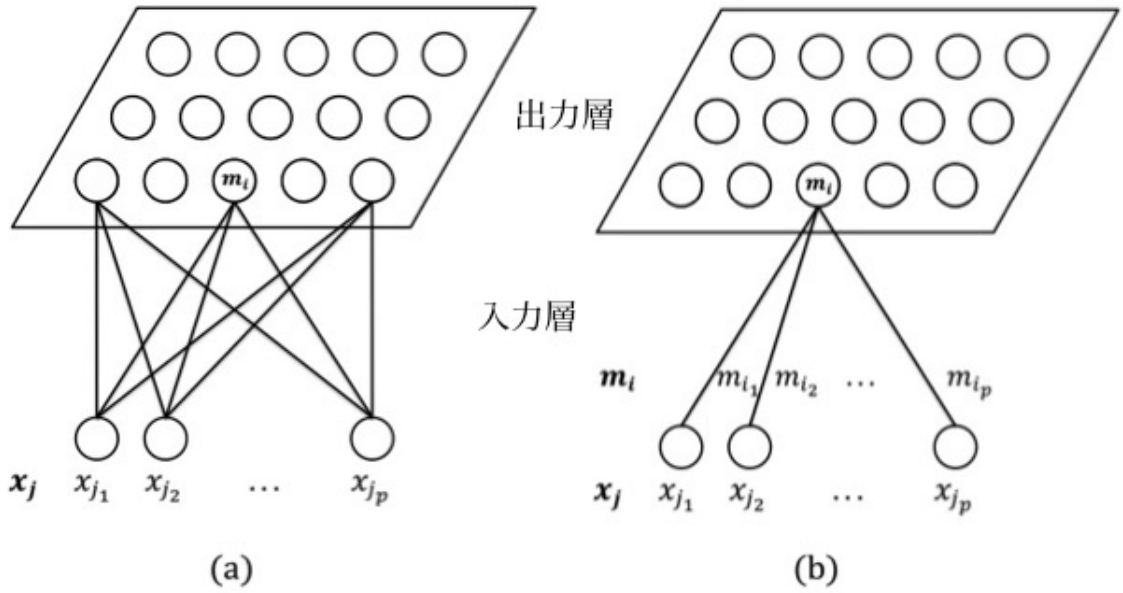

図 3.5: SOM の基本構造

第 4 章

ハニーポットによるデータ収集

本章ではハニーポットを用いて DRDoS 攻撃を観測し, データを収集するための手法について説明する. また, 収集したデータを分析して統計情報を示す.

4.1 DRDoS ハニーポット

攻撃者に脆弱なシステムであると見せかけることで攻撃を誘い込み, 攻撃を詳細に解析するシステムをハニーポットという.

このハニーポットの通信を観測することで DRDoS 攻撃を観測し，攻撃の分析を行う. ハニーポットはインターネット上に存在するリフレクタとは異なり正規の目的で利用するユーザは存在しないため，ハニーポットと通信を行うのは悪意を持った攻撃者や，セキュリティ関係の研究者のみである. そのため，正規の通信の中に攻撃が含まれる ISP バックボーン等のフローデータよりも高精度に DRDoS 攻撃を検知することが可能である. それらの攻撃トラヒックを分析することで，DRDoS 攻撃の特徴や傾向を把握できる. このように，ハニーポットは DRDoS 攻撃の観測や早期検知において有用な手法である.

本研究では DRDoS ハニーポットと呼ばれる, DRDoS 攻撃に利用される複数種類のネットワークサービスを提供するサーバを東日本, 西日本, 米国の 3ヶ所のインターネット上に設置した. それぞれのハニーポットで取得したデータを用いて分析を行う. この DRDoS ハニーポットの動作を描いた図を図 4.1 に示す.

図 4.1 のリフレクタは攻撃者からのスキャンに対してプロトコル非準拠で応答を行う. なお, 攻撃者は増幅率が大きいリフレクタを探索することが想定される [6] ため, 応答のパケットサイズを拡大して返信する. そして, 返信した IP アドレスに対して, 返信したポート番号で攻撃を待ち受けることで攻撃トラヒックを観測する. このとき攻撃者からの送信元 IP アドレスを詐称したパケットに対する応答は, 攻撃者に加担することになるため基本的には行わないが, この動作を見た攻撃者が振る舞いを変える可能性があるため, 随時に攻撃先にパケットを送信する.

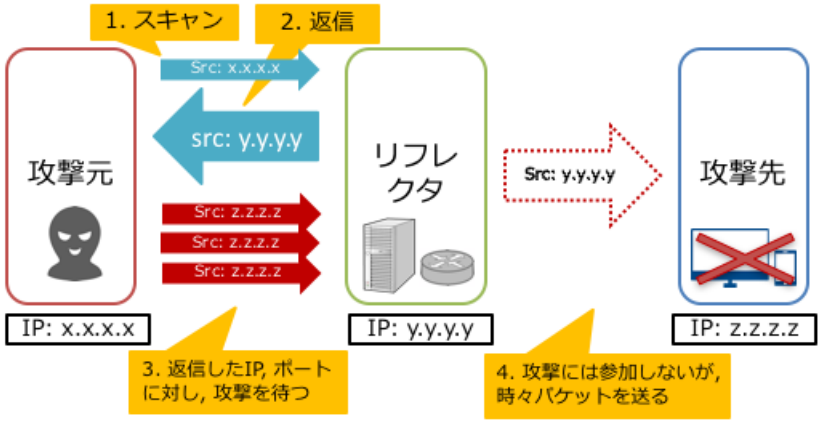

図 4.1: DRDoS ハニーポットの動作

4.2 収集したデータの分析

前述の DRDoS ハニーポットを用いて, 2015 年 12 月 22 日から 2016 年 1 月 31 日の間に攻撃通信を pcap ファイルに保存し, 分析を行った. 以下にその分析結果を示す.

4.2.1 ポート番号

DRDoS ハニーポットで観測された UDP パケットの宛先ポート番号を表 4.1 に示す. この表より, NTP を悪用した攻撃が大部分を占め, SSDP, chargen, RIP, SNMP を含めると約 99%を占めていることが確認できる. このことから, 本研究での検知の対象を, 表に記載したの 5 つのプロトコルとする.

表 4.1: 攻撃先のポート番号

UDP ポート番号	アプリケーション名	東日本 [%]	西日本 [%]	米国 [%]	合計 [%]
123	NTP	69.85	95.81	80.02	88.15
19	chargen	7.31	0.30	15.97	9.27
1900	SSDP	1.81	3.00	1.90	1.08
161	SNMP	4.43	0.07	0.23	0.14
520	RIP	12.71	0.002	0.25	0.12
その他	—	3.89	3.82	1.63	1.24

4.2.2 パケット数の推移

図 4.2, 図 4.3, 図 4.4 に, 各地点に設置された DRDoS ハニーポットへのパケット数の推移を示す. プロトコルによって規模は異なるが, 日々攻撃トラヒックが来続けていることが確認できる.

また, 計測開始から 504 時間後にハニーポットからの応答を停止しているが, 応答停止後もハニーポットへの攻撃が止まないことも確認できる.

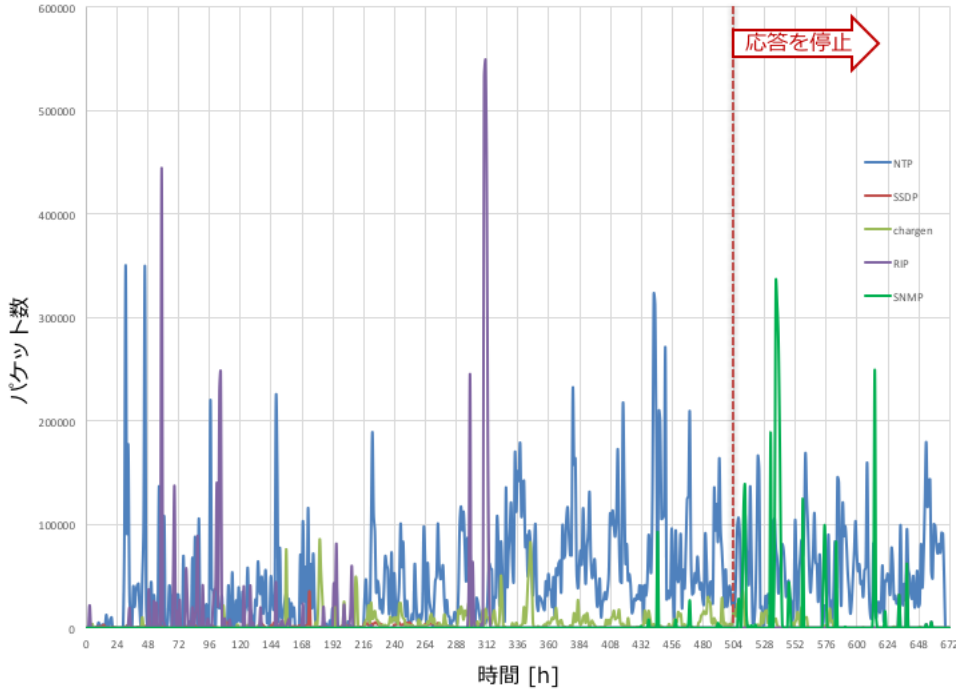

図 4.2: 東日本の DRDoS ハニーポットへのパケット数の推移

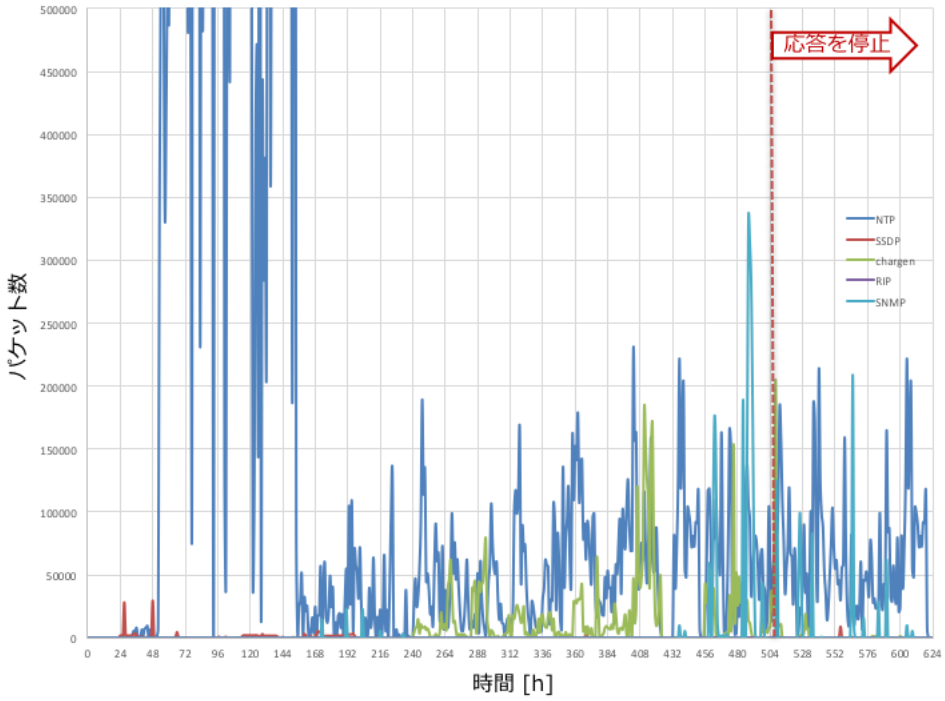

図 4.3: 西日本の DRDoS ハニーポットへのパケット数の推移

4.2.3 同一ホストへのパケット数の推移

図 4.5, 図 4.6 に, 各地点に設置された DRDoS ハニーポットの, 最も攻撃件数が多かったあるホストへの攻撃先パケット数の推移を示す. なお, 図 4.6 は図 4.5 の拡大図である. これらの図より, 同一の攻撃者からの攻撃が, 複数のリフレクタを用いて行われていることが確認でき, DRDoS 攻撃が観測できたといえる.

4.2.4 1 分間あたりの最大流量と総パケット数の相関関係

同一送信元 IP アドレスごとの, 1 分間あたりの最大流量と総パケット数の相関関係を図 4.7 に示す. この図より, 1 分間あたりの最大流量と総パケット数が大きい, 明らかな攻撃と思われる通信のほかに多くの通信が観測されることがわかる. 1 分間あたりの最大流量が大きいが総パケット数があまり多くないものはスキャンである可能性があるが, 攻撃通信との線引きが難しい. また, 1 分間あたりの最大流量が小さいが総パケット数が大きい通信はリフレクタによる攻撃の確認を行っているものと予想できるが, 4.2.2 項で示したように応答を完全に停止しても攻撃が止まなかったことから, 実態は精査する必要がある.

また, 図 4.7 のうち, 宛先 IP アドレスが TCP/HTTP 疎通ができたもののみを示した相関図を図 4.8 に示す. 1 分間あたりの最大流量と総パケット数が大きいの攻撃先は通信販売サイト

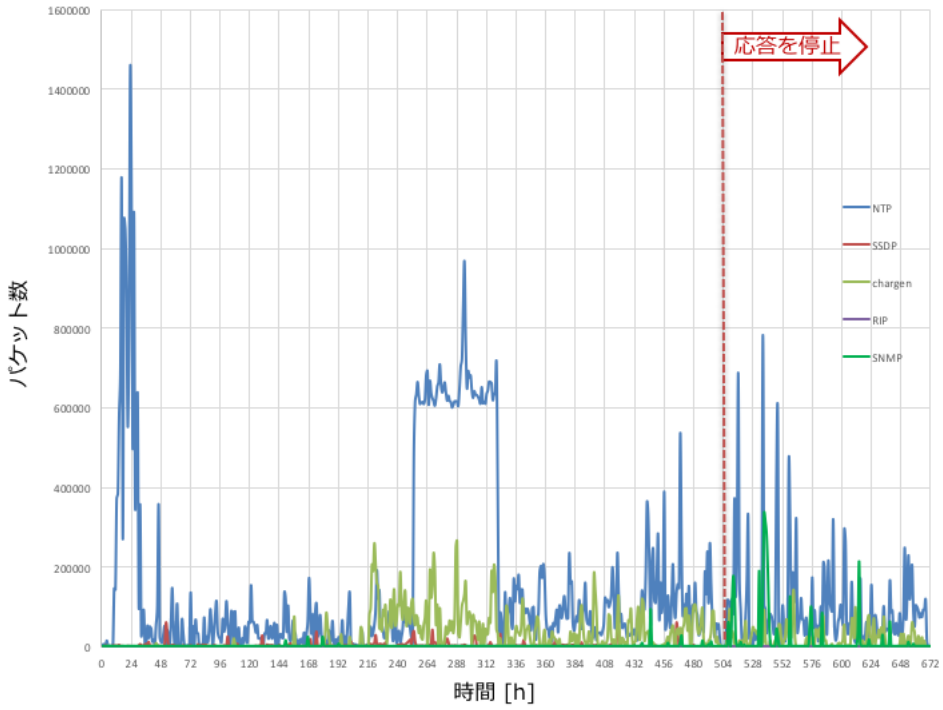

図 4.4: 米国の DRDoS ハニーポットへのパケット数の推移

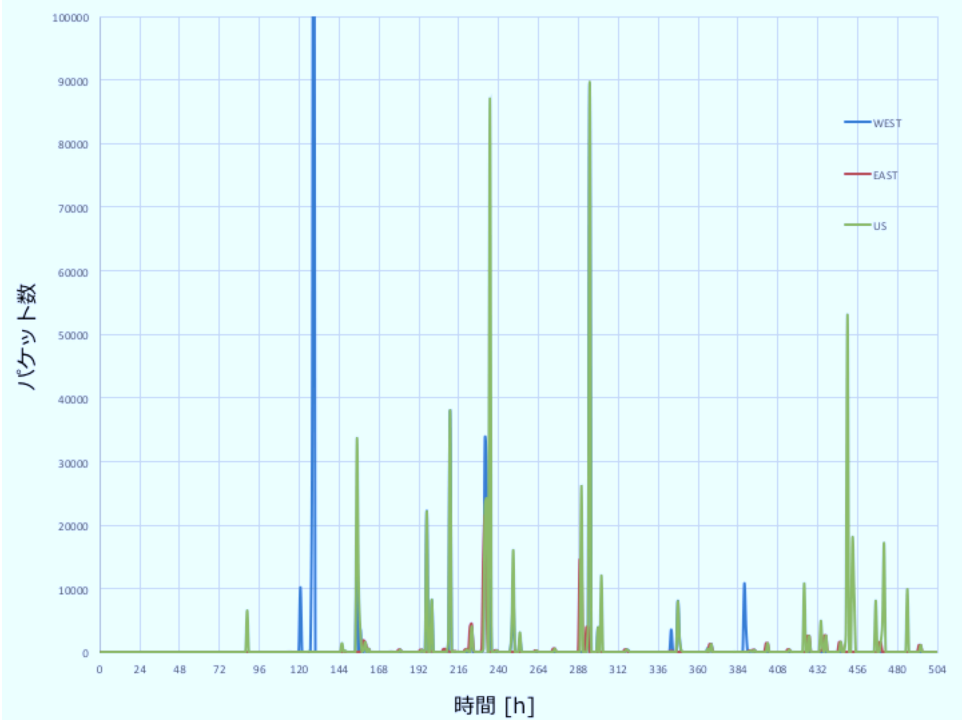

図 4.5: 同一ホストへの攻撃パケット数推移

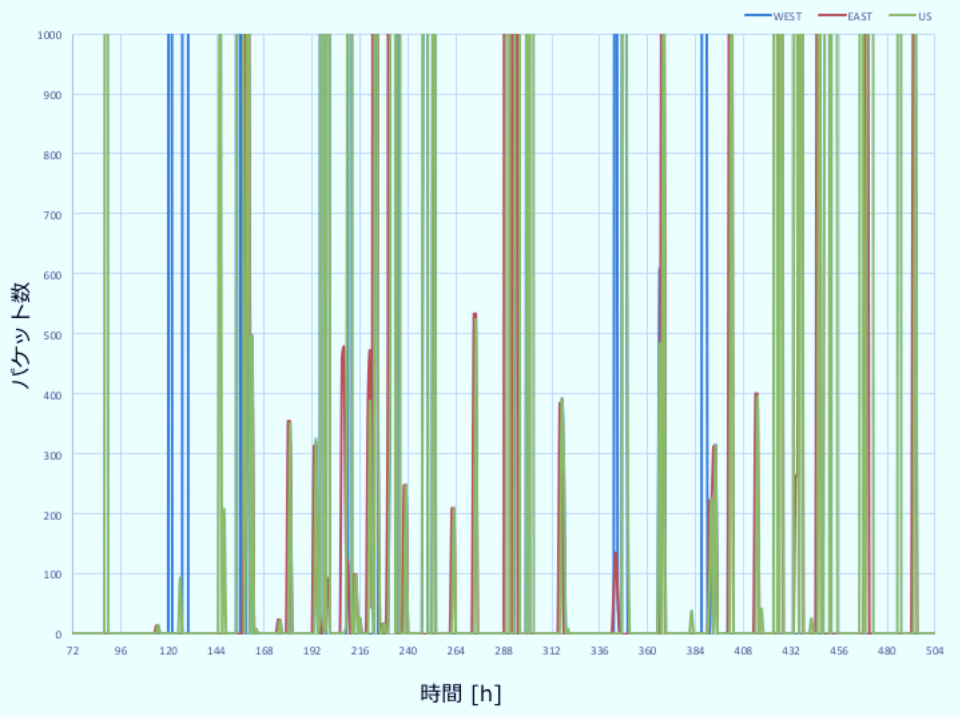

図 4.6: 同一ホストへの攻撃パケット数推移 (拡大図)

や教育機関のサイトが多かった. 1 分間あたりの最大流量が大きいが総パケット数があまり多くないものは ShadowServer [26] などの調査機関による通信が大部分を占めていた. さらに, 1 分間あたりの最大流量が小さいが総パケット数が大きい通信は IoT 端末のログイン画面に繋がる割合が高いことから, 攻撃者に利用されている IoT 端末が多数存在することが確認された.

4.2.5 攻撃の継続時間

3 つのハニーポットでの攻撃の継続時間を示したグラフを図 4.9 に示す. この図より, 60 の倍数や 100 の倍数といった切りのよい秒数の間続く攻撃が多いことがわかる. 前述した DDoS 代行サービスが増えていることによると考えられ, 攻撃トラヒックである可能性が高い.

そのため, 前項での分析結果も踏まえ, 本研究では特に割合の高かった, 100, 120, 180, 200, 300, 600, 1000, 1200, 1800 秒間続いたものを攻撃通信と見なし, 次章の実験データとして用いる.

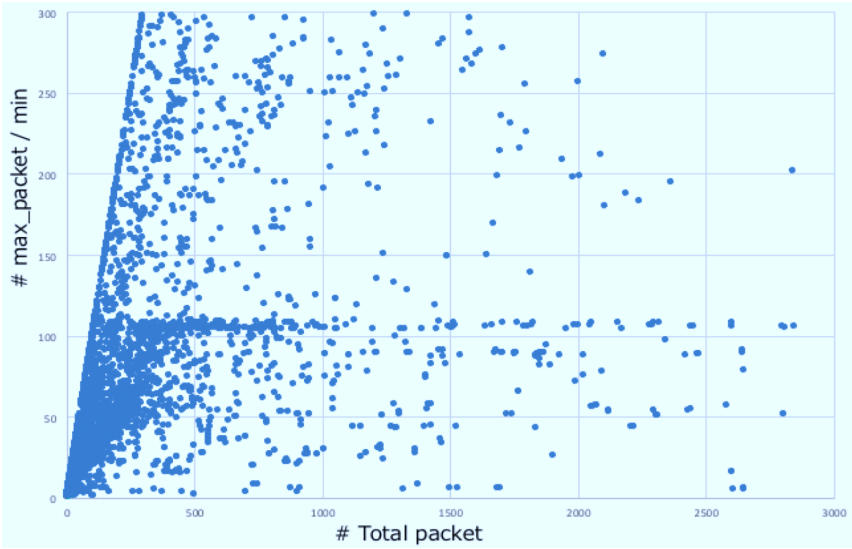

図 4.7: 1 分間あたりの最大流量と総パケット数の相関関係

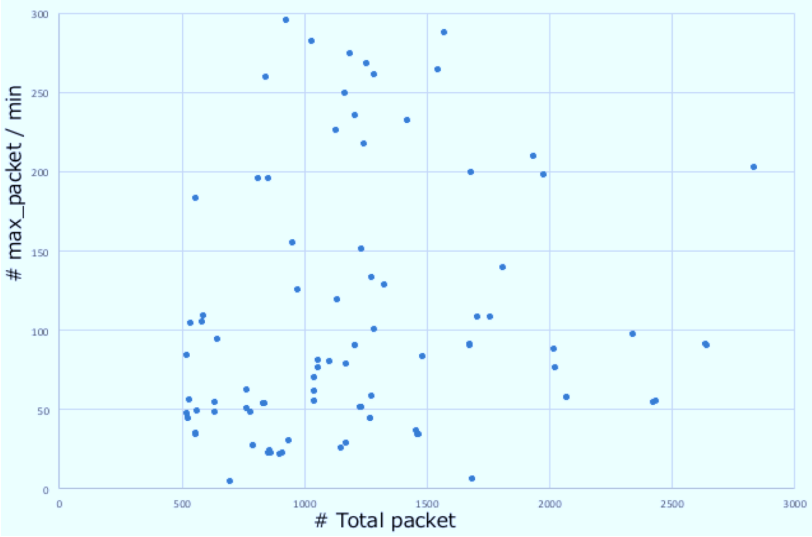

図 4.8: TCP/HTTP 疎通可能な送信元の 1 分間あたりの最大流量と総パケット数の相関関係

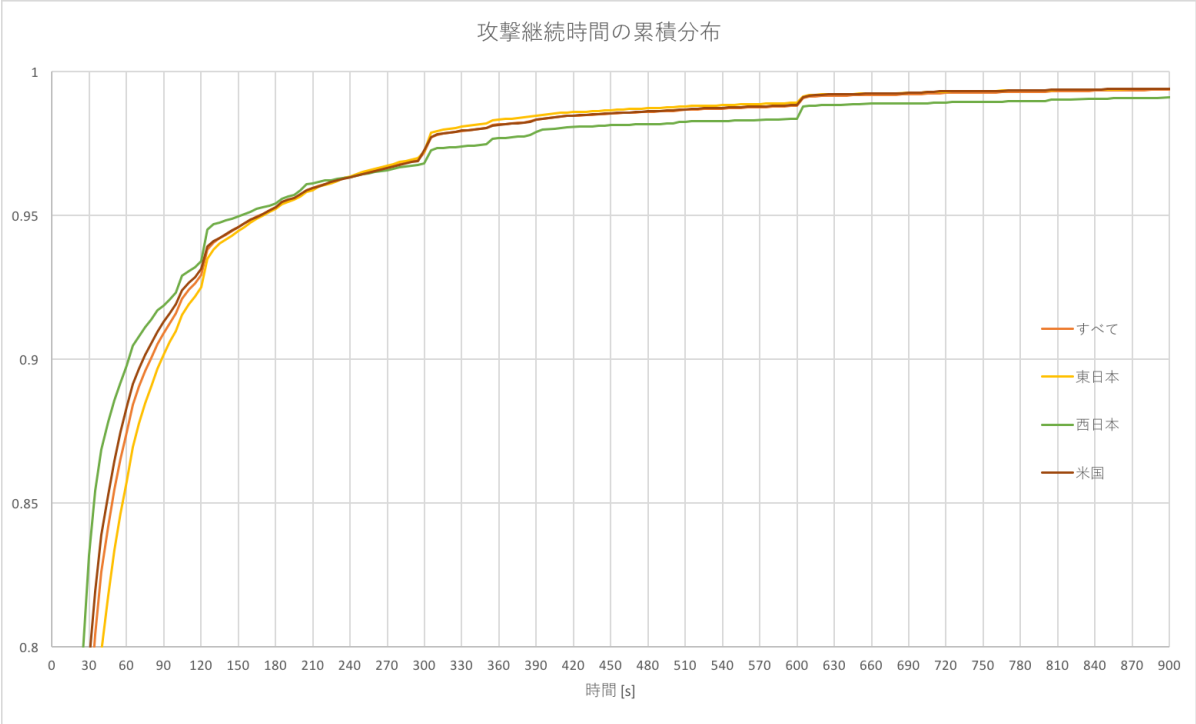
攻撃継続時間の累積分布
1
0.95
0.9
0.85
0.8
0 30 60 90 120 150 180 210 240 270 300 330 360 390 420 450 480 510 540 570 600 630 660 690 720 750 780 810 840 870 900
時間 [s]
すべて
東日本
西日本
米国

図 4.9: 攻撃の継続時間

第 5 章

提案手法

本研究では DRDoS ハニーポットへの通信トラヒックを分析し, パケットレベルで DRDoS 攻撃に用いられる通信の特徴を把握し, 機械学習により検知することを検討する.

本章では性能評価実験に用いるデータセット, 機械学習に用いる特徴量の決定方法, 提案手法の性能評価指標, 実験環境について順に述べた後, 本実験の処理の流れについて説明する.

5.1 データセット

データセットは正常通信のデータと不正通信のデータに分かれる. 不正通信のデータは, 前章で述べた 3 地点にある DRDoS ハニーポットを用いて 2015 年 12 月 22 日から 2016 年 1 月 31 日に収集した.

ハニーポットで取得した不正な通信と比較するための正常通信としては，大学で実運用中の/16 アドレスのネットワークの通信データを用いる. このデータは，大学のゲートウェイで計測されたトラフィックデータであり，2016 年 11 月 20 日から 11 月 25 日にかけて取得したキャプチャデータである. 本研究では，対象としている 5 つのプロトコルに絞ってデータを収集した.

5.2 特徴量

5.2.1 特徴量の決定方法

前節で述べたデータセットから特徴量を抽出し, 機械学習に使用する. なお, 抽出にはネットワークアナライザの wireshark[27] の CUI 版である tshark[28] を用いる.

一般的に多くの特徴量を用いれば精度の向上が期待できるが, その分計算量が増えるため処理時間が増大する. そのため, まず取得した通信データから tshark によって取得可能なすべてのフィンガープリント [29] を抽出する. IP は 132 種, UDP は 27 種, NTP は 122 種, RIP は 18 種, SNMP は 145 種, SSDP は 74 種あるフィンガープリントのうち, 統計をとることで正常通

信のデータと不正通信で差が見られると判断したものを, 特徴量の候補として用いる. 表 5.1 に特徴量の候補を示す. その後 J48 アルゴリズムによって作られる決定木に用いられている特徴量を検知に有用な特徴量として採用する.

表 5.1: 候補特徴量一覧

プロトコル名	特徴量
—	パケット長, パケットの到着間隔
IP	ヘッダ長, データ長, TTL, ID チェックサム, フラグ (Flags)
UDP	データ長, チェックサム 送信元ポート番号
NTP	LI (Leap Indicator), VN (Version Number), Mode Flags (LI, VN, Mode), 階層, ポーリング間隔, 精度 リクエストコード, ルート遅延, ルート拡散, 参照識別子 参照タイムスタンプ, 開始タイムスタンプ 受信タイムスタンプ, 送信タイムスタンプ
RIP	コマンド, バージョン メトリック, アドレスファミリ
SNMP	コミュニティ名, データ GetRequest, GetNextRequest GetResponse, GetBulkRequest Max Repetitions, Variable Bindings オブジェクト名, バージョン, タイムスタンプ
SSDP (HTTP)	リクエスト、ホスト Prev Request in frame
chargen	データ (文字列)

5.2.2 スケーリング

SVM や SOM を用いて判別する際は, 各特徴量は取り得る最小値と最大値 [30] を用いてスケーリングを行う. スケーリングとは, 最小値が −1, 最大値が 1 となるよう正規化することである. スケーリングを行うことによって, 数値の範囲の大きな特徴量が数値の範囲の小さい特徴量を支配してしまうことを防ぐことができる. また, 本研究で用いるガウシアンカーネルなど, 多くのカーネル関数では特徴量ベクトルの内積を用いて計算を行うため, 大きな値と小さな値の積の計算時に起こる情報落ちによる誤差の発生を防ぐことができる. これらの効果によって, 事前に特徴量のスケーリングを行うことが精度の向上につながると期待される.

5.3 性能評価の指標

本研究の提案手法の性能の評価指標について述べる. 本研究では評価指標として検知率, 誤検知率, 処理速度を用いる.

実際に不正なパケットを不正であると判別できた場合を TP (True Positive), 実際に不正なパケットを正常と判別してしまった場合を FN (False Negative), 実際に正常なパケットを不正と判別してしまった場合を FP (False Positive), 実際に正常なパケットを正常と判別できた場合を TN (True Negative) という. この関係を表 5.2 に示す. 検知率と誤検知率はこれらを用いて計算する. 検知率は式 (5.1) で計算される値であり, 実際に不正なパケットのうち, 不正であると判別できたパケットの割合である. 誤検知率は式 (5.2) で計算される値であり, 実際に正常なパケットのうち, 不正であると判別してしまったパケットの割合である.

処理速度は式 (5.3) で計算する.

表 5.2: 真の結果と判別結果の関係

		判別結果	
		malicious	benign
真の結果	malicious	TP (True Positive)	FN (False Negative)
	benign	FP (False Positive)	TN (True Negative)

$$\text{検知率} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}} \tag{5.1}$$

$$\text{誤検知率} = \frac{\mathbf{FP}}{\mathbf{FP} + \mathbf{TN}} \tag{5.2}$$

$$処理速度 = \frac{パケット長の合計}{特徴量抽出から判別までの時間} \tag{5.3}$$

5.4 実験環境

本実験を行う環境を表 5.3 に示す.

表 5.3: 実験環境

OS	Ubuntu 14.04 LTS 64bit
CPU	Intel Xeon CPU E3-1220 V2 (8M Cache, 3.10GHz)
RAM	8GB

5.5 提案手法の手順

提案手法の手順を以下に示す. まずパケットから検知に用いる 5.1 節において述べた, 特徴量の候補となるなるべく多くの情報を抽出する. 次に, それらの統計分析により正常通信データとハニーポットへの攻撃通信で差が見られたものを選んだ後, J48 アルゴリズムを用いて機械学習手法の一つである決定木を作成する. これによって自動的に判別に有用な特徴量が絞り込まれる. こうして決定された特徴量を用いて, 複数の機械学習手法を比較することで, 検知に最適な機械学習法を明らかにする. なお実験において交差検定は行わず, 時系列に並ぶデータの前半部分を学習用データ, 後半部分をテスト用データとして用いる. 3ヶ所のハニーポットで取得したそれぞれの DRDoS 攻撃データに対して, 同一の正常データを用いて実験を行う.

実験手順

1. 候補となる特徴量の抽出と統計分析
2. J48 を用いた決定木を構築による特徴量ベクトルの決定
3. 決定した特徴量ベクトルを用いた, 機械学習法の比較

第 6 章

実験結果と考察

ここでは, 提案手法による実験を行った結果とそれらに対する考察を行う.

6.1 実験結果

6.1.1 判別に有効な特徴量

最初に決定木の構築アルゴリズムである J48 を用いて決定木を作成することで数が絞り込まれた特徴量を表 6.1 に示す.

表 6.1: 決定木作成に用いられた特徴量

プロトコル名	東日本	西日本	米国
—	パケット長	パケット長, 到着間隔	パケット長, 到着間隔
IP	TTL	—	TTL
UDP	送信元ポート番号	送信元ポート番号	送信元ポート番号
NTP	Flags (LI, VN, Mode)	Flags (LI, VN, Mode)	Flags (LI, VN, Mode) リクエストコード
RIP	—	—	—
SNMP	Max Repetitions	—	GetBulkRequest
SSDP	—	—	—
chargen	—	—	—

6.1.2 判別器の比較

前節で決定した特徴量を用いて, 決定木と他の機械学習法であるランダムフォレスト (RF), SVM, SOM, を用いた場合の性能を比較した. その結果を表 6.2 に示す. なお, 結果は小数第 3 位以下を検知率については切り捨て, 誤検知率については四捨五入して示す. また, ここで用いた不正通信は東日本に設置された DRDoS ハニーポットで収集されたものである.

表 6.2: 機械学習法の比較

	決定木	RF	SVM	SOM
検知率 [%]	99.99	100.00	98.71	93.63
誤検知率 [%]	0.01	0.01	0.45	1.34
判別速度 [Mbps]	293.95	292.21	43.66	36.92

6.2 考察

判別に有効な特徴量は下位層のプロトコルのものが多いということがわかった. 特にパケット長や, IP ヘッダの TTL に関してはほとんどの実験において使用されていた. そこで, それぞれのハニーポットにおける, パケット長の累積分布を図 6.1, 図 6.2, 図 6.3, TTL の累積分布を図 6.4, 図 6.5, 図 6.6 にそれぞれ示す. これらの図から, DRDoS 攻撃に用いられるパケットのパケット長や TTL はある値に偏る場合が多く, 分類する際の特徴量として有効であることがわかる. なお, 図 6.5 より, 西日本に設置したハニーポットの TTL は偏っておらず, そのため特徴量として用いられていないことが, 表 6.1 より確認できる. これは, パケット長に関しては, 攻撃者はなるべく増幅率が高くなるパケットを送るため, 同じようなパケットが送信されるためと考えられる. また TTL に関しては, 攻撃者は専用のツールを用いて攻撃を行うため, 一般的ではない値に偏るためと考えられる. 特に本研究では DDoS 攻撃代行サービスが行っている可能性の高いデータを使用しているため特徴が顕著に出ると考えられる.

次に各プロトコルについて考察をする. 表 6.1 より, NTP において，flag が特徴量として使われている. これは VN フィードが 0, 1, 2 のものは古いバージョン を狙ったものであり，また, Mode フィールドが 6 や 7 のものは通常使われないが問い合わせに対する応答サイズが大きいため，増幅率が高くなることが原因として考えられる. また, リクエストコードに関しては, monlist コマンドによる過去の履歴要求が攻撃に利用されるためと考えられる. また, SNMP については GetBulkRequest のフラグが用いられている. これも増幅率を高くするためであり, 米国のハニーポットでは約 93.0%が GetBulkRequest であった. 一方, chargen では増幅率を高く

するためにデータ部が 0 バイトにされるという傾向や, RIP はバージョン 1 が狙われるという特徴があるが, 本実験ではそれらを用いなくとも, 高い判別率を有していた.

学習器を比較すると, ランダムフォレストによる判別率が最も高く, 判別速度も非常に高速であった. また, SVM や SOM はマルウェアの活動等の不正通信の判別には非常に適している [31] が, 本研究では, 若干劣っており, 特に判別速度が遅いため DRDoS の検知にはランダムフォレストや決定木の方が適している.

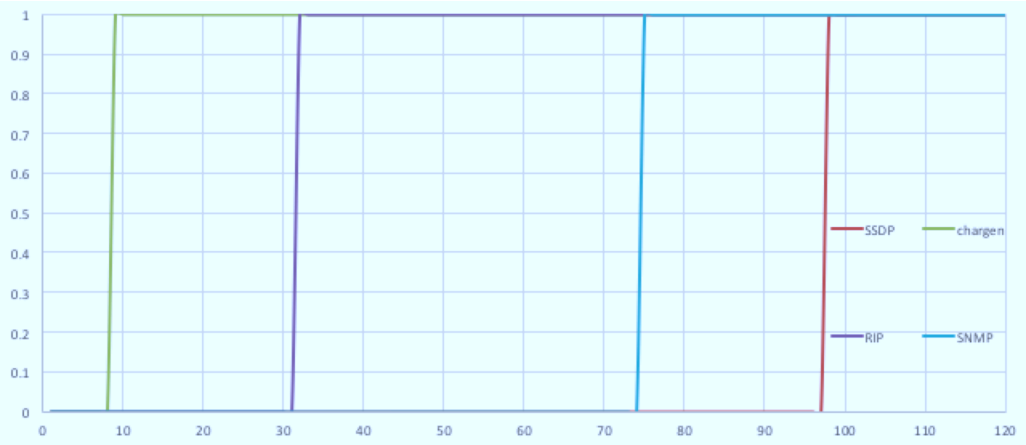

図 6.1: 東日本の DRDoS ハニーポットのパケット長の累積分布

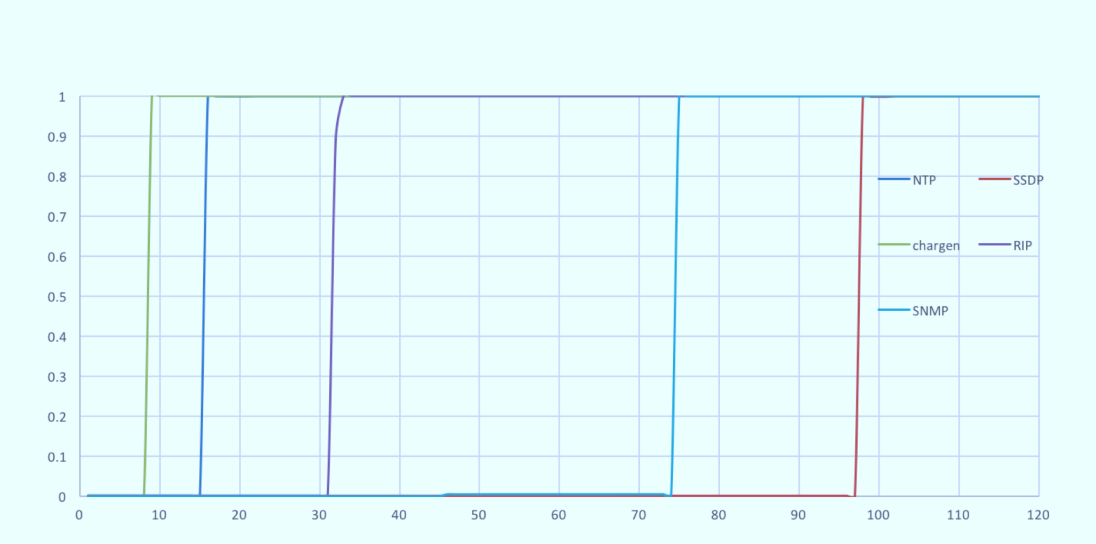

図 6.2: 西日本の DRDoS ハニーポットのパケット長の累積分布

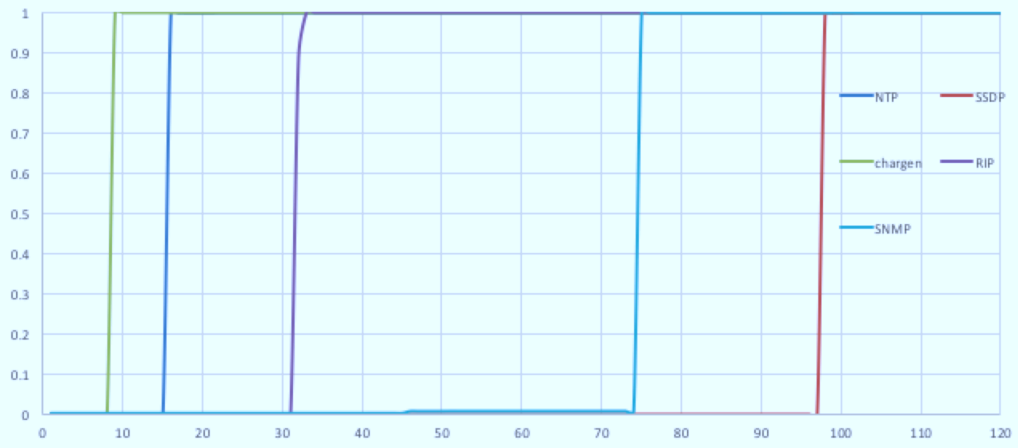

図 6.3: 米国の DRDoS ハニーポットのパケット長の累積分布

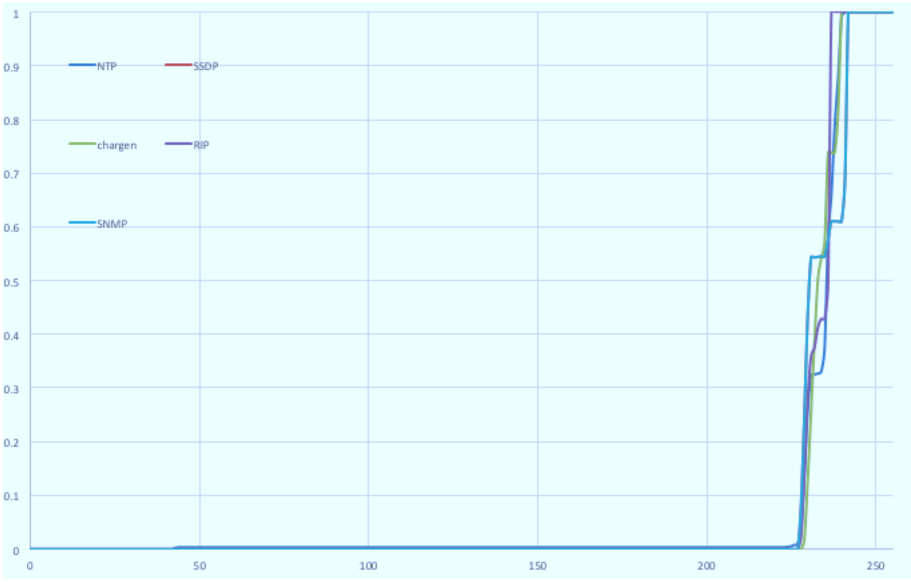

図 6.4: 東日本の DRDoS ハニーポットの TTL の累積分布

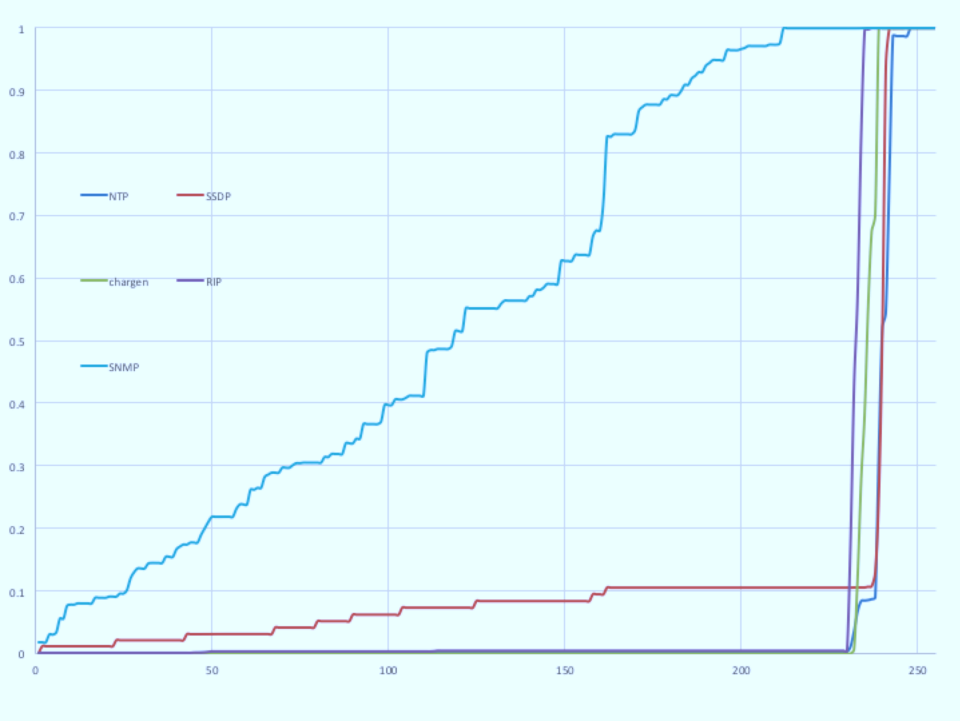

図 6.5: 西日本の DRDoS ハニーポットの TTL の累積分布

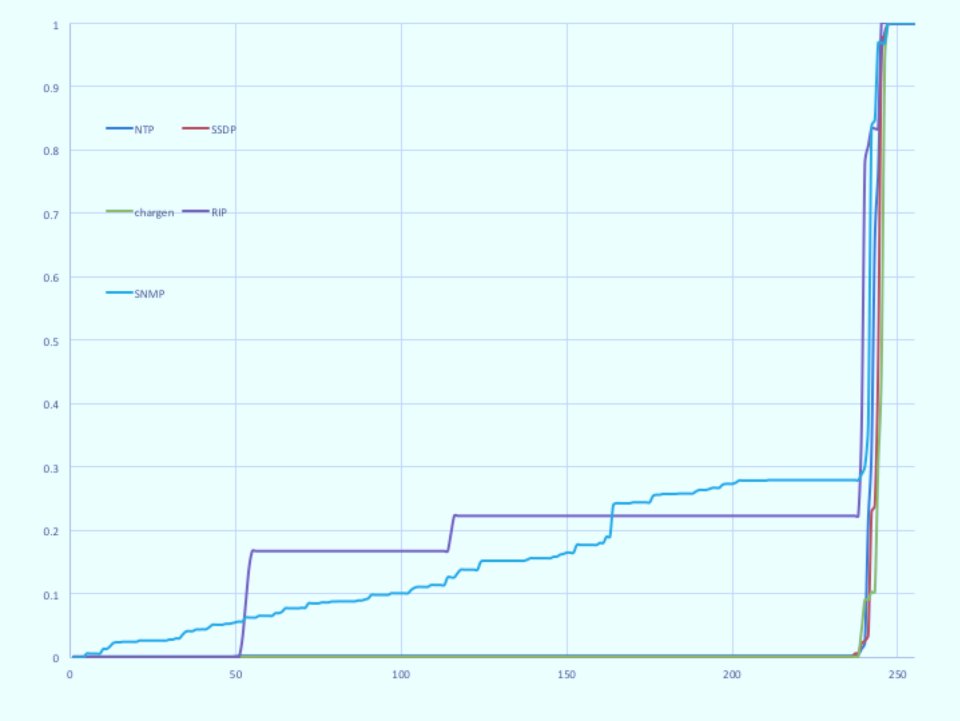

図 6.6: 米国の DRDoS ハニーポットの TTL の累積分布

第 7 章

まとめと今後の課題

7.1 まとめ

日米の 3ヶ所に設置された, DRDoS 攻撃に悪用されるリフレクタを模擬するシステムである DRDoS ハニーポットを用いて DRDoS 攻撃の傾向を調査した. 攻撃の種類や, リフレクタの応答停止によるパケット数の推移, 複数のリフレクタを同時に使った攻撃の観測, 攻撃の継続時間, 攻撃以外の通信の存在, さらに IoT 端末による攻撃を確認した.

次に, DRDoS 攻撃をパケットレベルで高精度に判別するために有効な特徴量を機械学習によって自動的に選別できることを明らかにした. 選ばれた特徴は攻撃者が増幅率を高くするために用いられるものであった. また, 判別に適した機械学習法についても実験によって明らかにした. その結果, その結果, ランダムフォレストが最も高精度に正常通信との判別が行えることを示した. 提案手法による実験を行った結果, 性能指標は検知率 100.0%, 誤検知率 0.01%と非常に精度が高く, 処理速度も実験環境において約 300Mbps であり, 高精度かつ高速な判別が可能であることを示した.

7.2 今後の課題

DRDoS 攻撃は短時間の間に多量のパケットが送られるため, ハニーポットに来るすべてのパケットを判別することは困難である. よって攻撃を見逃さないようしつつ, サンプリングする手法を考える必要がある.

また, DRDoS ハニーポットには DoS 攻撃と考えられる通信以外にもスキャンのほか, 実体の不明な通信が多く含まれ, 攻撃者が悪意を持って攻撃している通信のみを切り出すのが困難であることを第 4 章において述べた. そのため, DRDoS ハニーポットに来る通信を詳細に分析し, ハニーポットへの通信のうち攻撃通信のみを分類する方法を明らかにすることで, より精度の良い分類が可能になると考えられる.

さらに, 本研究では有効な特徴量の決定法について, 決定木を用いるもの以外の方法を検討していない. また, 判別に用いる学習器についてはセキュリティ分野でよく用いられる 4 種類の方法についてしか検討してない. 今日注目されているディープラーニングなど, 特徴量の決定や判別が可能な機械学習法は他にもあるため, それらについても検討することで, より性能の良い検知ができる可能性がある.

謝辞

本修士論文の作成にあたり, 日頃より御指導をいただいた早稲田大学 基幹理工学研究科 情報理工・情報通信専攻の後藤滋樹教授に深く感謝致します.

また, NTT セキュアプラットフォーム研究所の神谷和憲氏には本研究におけるデータの提供や研究のアドバイス等で多大なご協力をいただきました. 深くお礼申し上げます.

最後に, 日ごろお世話になった後藤滋樹研究室の皆様に感謝致します.

参考文献

[1] ARBOR Networks, “Worldeide Infrastructure Security Report Volume XI,” https://www.arbornetworks.com/images/documents/WISR2016_EN_Web.pdf, 2016.

[2] Akamai, “Q2 2016 State of the Internet Security Report,” https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/akamai-q2-2016-state-of-the-internet-security-report.pdf, 2016.

[3] 牧田大佑, 吉岡克成, “DRDoS 攻撃を観測するハニーポット技術の研究開発,” http://www.nict.go.jp/publication/shuppan/kihou-journal/houkoku-vol62no2/K2016S-05-03.pdf , 情報通信研究機構研究報告 Vol. 62 No. 2, 2016.

[4] US-CERT, “UDP-Based Amplification Attacks,” https://www.us-cert.gov/ncas/alerts/TA14-017A, 2016.

[5] 牧田大佑, 西添友美, 小出駿, 筒見拓也, 金井文宏, 森博志, 吉岡克成, 松本勉, “早期対応を目的とした統合型 DRDoS 攻撃観測システムの構築,” Symposium on Cryptography and Information Security 2015.

[6] 西添友美, 牧田大佑, 吉岡克成, 松本勉, “プロトコル非準拠のハニーポットによる DRDoS 攻撃の観測,” Symposium on Cryptography and Information Security 2016.

[7] 牧田大佑, 西添友美, 吉岡克成, 松本勉, 井上 大介, 中尾 康二, “早期インシデント対応を目的とした DRDoS 攻撃アラートシステム,” 情報処理学会論文誌 Vol.57 No.9 197420131985, Sep. 2016.

[8] 蒲谷 武正, 千賀 渉, 村上 洸介, 牧田 大佑, 吉岡 克成, 中尾 康二, “AmpPot を活用した DRDoS 攻撃対応早期化の取り組み,” Computer Security Symposium 2016.

[9] 浦川順平, 澤谷雪子, 山田明, 窪田歩, 牧田大祐, 吉岡克成, 松本勉, “ハニーポット監視による DRDoS 攻撃の早期規模推定,” Symposium on Cryptography and Information Security 2015.

[10] 柴原健一, 筒見拓也, 小出駿, 森博志, 村上洸介, 中尾康二, 吉岡克成, 松本勉, “DRDoS 攻撃を観測可能なダークネットを用いたリフレクタの分析,” Symposium on Cryptography and Information Security 2016.

[11] 牧田 大佑, 吉岡克成, 松本勉, 中里純二, 島村隼平, 井上大介, “DNS アンプ攻撃の事前対策へ向けた DNS ハニーポットとダークネットの相関分析,” , 情報処理学会論文誌 Vol.56 No.3 9212013931, Mar. 2015.

[12] Jose Jair Santanna, Roland van Rijswijk-Deij, Rick Hofstede, Anna Sperotto, Mark Wierbosch, Lisandro Zambenedetti Granville, Aiko Pras, “Booters - An Analysis of DDoS-as-a-Service Attacks,” http://dl.ifip.org/db/conf/im/im2015/137274.pdf, 2015.

[13] M. Karami, Y. Park, and D. Mccoy, “Stress Testing the Booters:Understanding and Undermining the Business of DDoS Services,” Proc.World Wide Web Conf., 2016.

[14] 鈴木崇広, “決定木によるデータマイニングの比較,” http://www.ccn.yamanashi.ac.jp/~munehisa/kenkyuu/soturon_2004/suzuki.pdf, 学士学位論文, 山梨大学, Feb. 2005.

[15] Machine Learning Group at the University of Waikato, “Weka 3: Data Mining Software in Java,” http://www.cs.waikato.ac.nz/ml/index.html, 2017 年 1 月 30 日閲覧.

[16] C. M. ビショップ, 元田浩, 栗田多喜夫, 樋口知之, 松本裕治, 村田昇 (監訳), “パターン認識と機械学習 上・下,” 丸善出版, 東京, 2008.

[17] 麻生英樹, 津田宏治, 村田昇, 甘利俊一, 竹内啓, 竹村彰通, 伊庭幸人 (編), “統計科学のフロンティア 6 パターン認識と学習の統計学 新しい概念と手法,” 岩波書店, 東京, 2005.

[18] 辻谷將明, 竹澤邦夫, 金明哲 (編), “R で学ぶデータサイエンス 6 マシンラーニング,” 共立出版, 東京, 2011.

[19] Yoshihiko Suhara, LinkedIn Corporation, “SVM 実践ガイド,” http://www.slideshare.net/sleepy_yoshi/svm-13435949, June 2012.

[20] Chih-Chung Chang, Chih-Jen Lin, “LIBSVM – A Library for Support Vector Machine,” http://www.csie.ntu.edu.tw/~cjlin/libsvm/, 2017 年 1 月 30 日閲覧.

[21] Thorsten Joachims, “SVM light,” http://www.cs.cornell.edu/People/tj/svm_light/index.html, Cornell University, Aug. 2008.

[22] r-project.org, “The R Project for Statistical Computing,” http://www.r-project.org/, 2017 年 1 月 30 日閲覧.

[23] Chih-Chung Chang, Chih-Jen Lin, “LIBSVM : A Library for Support Vector Machines,” http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf, 2017年1月30日閲覧.

[24] 金明哲, “Rによるデータサイエンス –データ解析の基礎から最新手法まで–,” 森北出版, 東京, 2007.

[25] Ron Wehrens, “Package ‘kohonen’,” http://cran.r-project.org/web/packages/kohonen/kohonen.pdf, Sep. 2015.

[26] Shadowserver, shadowserver, https://www.shadowserver.org/wiki/, 2017年1月30日閲覧.

[27] wireshark.org, “Wireshark,” https://www.wireshark.org/, 2017年1月30日閲覧.

[28] wireshark.org, “tshark - The Wireshark Network Analyzer 1.12.2,” https://www.wireshark.org/docs/man-pages/tshark.html, Jan. 15, 2015.

[29] Wireshark.org, “Wireshark Display Filter Reference”, https://www.wireshark.org/docs/dfref/#section_s, 2017年1月30日閲覧.

[30] W・リチャード・スティーブンス, 橘康雄 (訳), 井上尚司 (監訳), “詳解TCP/IP Vol.1 プロトコル,” ピアソン桐原, 東京, 2012.

[31] Kazuma Shinomoiya, Shigeki Goto, ”Detecting Malicious Traffic through Two-phase Machine Learning,” Poceedings of the Asia-Pacific Advanced Network, vol.40, pp. 34-40, Oct. 2015.