DDoS攻撃によるバックスキャッタ検出に関する研究

A study on Detection System of Backscatter due to DDoS Attack

栗原斗南，山﨑浩一

Tonan Kurihara and Koichi Yamazaki

玉川大学工学部ソフトウェアサイエンス学科，　194-8610　東京都町田市玉川学園6-1-1

Department of Software Science, College of Engineering, Tamagawa University,
6-1-1 Tamagawagakuen Machida-shi Tokyo 194-8610

Abstract

In this paper, we consider the machine learning system, which was proposed in a preceding work, that detects Distributed Denial of Service (DDoS) backscatter packets from those observed in the darknet. We analyze the effect of 14 feature values on the precision and the learning time of the system. Packets for 80/TCP are used for this study. It is shown that some feature values improve precision of backscatter detection at the cost of the learning speed, and the other increase the learning speed with little loss of precision.

Keywords: DDoS attacks, Backscatters and Darknet

1. はじめに

インターネットを利用している様々なサービスにとってDDoS(Distributed Denial of Service)攻撃は深刻な損害を与える脅威の1つである．まず，サーバの容量を超えた大量アクセスを引き起こすことでサーバがパンクすることを利用し，悪意を持ってサーバに大量のデータを送り付けるサイバー攻撃のことをDoS(Denial of Service)攻撃と呼ぶ．本論文で扱うDDoS攻撃とは，DoS攻撃を発展させ，複数のIPアドレスから分散してDoS攻撃を行うサイバー攻撃である．DDoS攻撃は，同一のIPアドレスからのアクセス回数を制限することで防御することができるDoS攻撃と異なり，マルウェアなどで不正に乗っ取った複数のIPアドレスを踏み台にしてDoS攻撃を行うことから，以下の2つの問題が挙げられる．

・攻撃元のIPアドレスが複数あるため，選択してその特定のIPアドレスをブロックすることが難しい

・マルウェアなどで不正に乗っ取ったIPアドレスを踏み台にしているため，攻撃者を割り出すことが難しい

DDoS攻撃の被害事例として，2016年秋以降に蔓延したマルウェア「Mirai」によって引き起こされた事件が挙げられる．この事件の一環で，大手DNSサービスDynが標的とされ，Dynのサーバに多大な負荷がかかり，Reddit，Twitter，Spotifyといった大手Webサービスの利用が一時困難になった．また，カスペルスキーの2020年第1四半期，第2四半期のレポートによると，2020年度はDDoS攻撃数が前年同期比の3倍になっているという調査結果が出ている[1,2]．これは，新型コロナウィルス感染症の世界的な大流行で外出機会が減り，オンラインサービスへの依存度が非常に高まっている現状を，サイバー攻撃者が悪用していることが理由の一つだと考えられている．これらのような事態を防ぐために，DDoS攻撃を素早く発見し，サービスの利用に支障が出る前に対応することが課題と

なっている．

サーバはどの通信が攻撃によるものなのか判別できないため，すべての通信に対して返答を行う．この返答のうち，DDoS攻撃によって発生したものをバックスキャッタと呼ぶ．インターネット上で到達可能かつ未使用IPアドレス空間であるダークネットからバックスキャッタを観測し，DDoS攻撃の特徴を分析することで，早期にDDoS攻撃を発見し，サービスが機能不全を起こす前に対策を行うことができる．ダークネットから観測を行う利点として，ダークネットに対して通常の通信によるパケットが送信されることは非常に稀であり，大部分をマルウェアが脆弱性を攻撃するためのパケットや，送信元IPアドレスが詐称されたパケットへの応答パケットなど，不正な活動を目的としたパケットが占めているため，異常な通信が多く，検出が容易だという点が挙げられる．先行研究として，DDoSバックスキャッタ検出システム[3]がある．この研究では，ダークネットで観測されたパケットを用意し，前処理，特徴生成を行うことで学習に用いるデータを作成している．その後，学習器にデータを入力し学習させることで，届いたパケットがバックスキャッタであるか非バックスキャッタであるかが判別できるシステムを作成している．本研究では，先行研究と同様のシステムを構成し，学習に使用する特徴ベクトルの組み合わせの変更によるシステムの特性への影響を解析する．その結果，特徴ベクトルがバックスキャッタ検出の精度及び学習時間におよぼす影響を明らかにする．

2. システム構築

本研究では先行研究[3]を参考に，ダークネットに届いたパケットからDDoS攻撃によるバックスキャッタを判別するシステムを構築する．

2-1. システム構築

今回構築したシステムの概要図を図1に示す．本システムは，まずダークネットから届いたパケットに処理を加え14個の特徴ベクトルを抽出する．続いて特定のパケットを教師データとして扱うため，ラベリングを行い，それ以外のパケットをテストデータとする．その後ラベリングを行った教師データを用いて，学習器に初期学習を行う．そして初期学習を行った学習器に対して，テストデータが，バックスキャッタであるか，非バックスキャッタであるかを判定する．

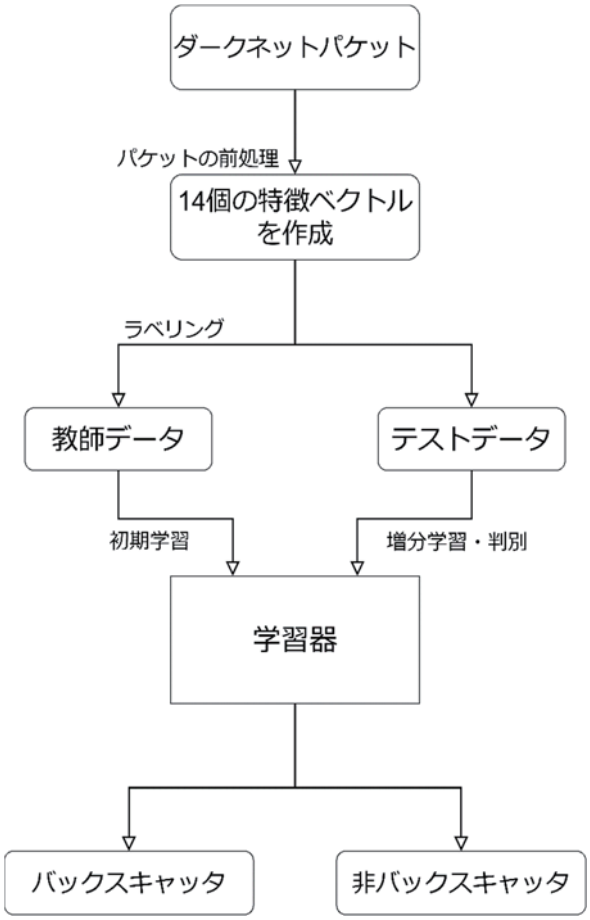

図1　提案システム概要図

今回扱うダークネットのパケットデータは，情報通信研究機構（NICT）で収集されているダークネットトラフィックデータを利用する[4,5]．このデータは公開することのできない情報であるため，提供された仮想環境にて開発を行う．

2-2. パケットの前処理及び特徴ベクトル作成

本節では，ダークネットから観測されたパケットから特徴ベクトルを作成するための前処理を

行い，特徴ベクトルの作成を行うまでの手順を説明する．前処理の手順は以下のとおりである．

1. パケットをIPアドレスでソートし，IPアドレスごとに以下の処理を行う．
2. 最初の1分間のパケットのデータを抽出する．
3. (2)で抽出したデータが10パケット未満だった場合，次のパケットより1分間のデータを新たに抽出する．これを10パケット以上になるまで繰り返す．
4. パケットの間隔が1時間空いた場合，新たに1分間のデータを抽出する．
5. 得られたデータから14個の特徴ベクトルを作成する．

この手順を行う根拠として，最初の1分のパケットからデータを抽出することで，DDoS攻撃の早期の発見ができると考える．また，1時間パケットの間隔があいた場合は，別の攻撃が行われていると考え，新たなデータの抽出を開始する．

14個の特徴ベクトルは，先行研究を参考に，TCPパケット，UDPパケットのどちらにも含まれる情報を用いる．14個の特徴は以下のとおりである．

・ パケット数
・ パケット間の時間差の平均値
・ パケット間の時間差の分散
・ 送信元ポートの数
・ 送信元ポートへのパケット数の平均値
・ 送信元ポートへのパケット数の分散
・ 送信先IPアドレスの数
・ 送信先IPアドレス へのパケット数の平均値
・ 送信先IPアドレス へのパケット数の分散
・ 送信先ポートの数
・ 送信先ポートへのパケット数の平均値
・ 送信先ポートへのパケット数の分散
・ ペイロードサイズの平均
・ ペイロードサイズの分散

その後，各特徴ベクトルの最大値が1，最小値が0となるように正規化を行う．これにより，すべての特徴を平等に判定することが可能になる．

2-3. ラベリング

本システムを作るうえで，まず学習器の初期学習を行うための教師データが必要になる．教師データの作成には，入力として用いる特徴ベクトルのペアとなる答えが必要になる．本節では，特徴ベクトルの答えである，バックスキャッタであるか非バックスキャッタであるかを特徴ベクトルにラベリングしていく．ラベリングを行うためには，バックスキャッタであるか非バックスキャッタであるか判別が可能であるパケットが必要になる．

2-3-1. TCPパケットのラベリング

ダークネットに届くバックスキャッタの判別が可能であるパケットとして，TCPでは3wayハンドシェイクの返信パケットを用いる．3wayハンドシェイクでは，TCP制御フラグにあるSYN，ACK，RSTなどの値を確認しながら行われる．まず3wayハンドシェイクの手順を以下に示す．

1. ホストAからホストBに接続確立を要求(SYN = 1)する．
2. ホストBがホストAからの接続要求に応え(ACK = 1)，ホストB からホストAに接続確立の要求(SYN = 1)をする．ここで，ホストAからの接続要求を断る場合はACKフラグの代わりにRSTフラグが立ったパケットが返される(RST =1, ACK = 0 or RST = 1, ACK = 1)．
3. ホストAがホストBからの接続要求に応える(ACK = 1)

ダークネットでは，通常は通信が行われることがないため，上記手順の2の工程で行っているような，接続要求に応える返信パケットや断るパケットがダークネットに送信されることがない．そのため，ダークネットで観測された返信パケットはDDoS攻撃によるバックスキャッタである可能性が高い．ここで，3wayハンドシェイクを用いて接続要求を行った場合，返されるパケットは以下の3通りである．

・ TCP制御フラグが「SYN = 1, ACK = 1」となっ

ているTCPパケット
・TCP制御フラグが「RST = 1, ACK = 0」となっているTCPパケット
・TCP制御フラグが「RST = 1, ACK = 1」となっているTCPパケット

3wayハンドシェイクが正しく行われると断定できるのはHTTPで用いられる80番ポートからのパケットである．そのためTCP80番ポートから送信されたパケットの内，TCP制御フラグがSYN-ACK，RST，RST-ACKのいずれかであるものをバックスキャッタであるとしてラベリングを行う．

2-3-2. UDPパケットのラベリング

UDPパケットにおいても，2-3-1で述べたTCPパケットと同様に，返信パケットが存在するデータをバックスキャッタと判定する目印とする．そこで，UDP通信で行われるDoS攻撃の内，返信パケットが存在するものとして，DNSサーバを狙うDoS攻撃が挙げられる．そのため，ここではUDPパケットの中でも特にDNSパケットに注目する．ペイロード内部の文字列を確認し，ドメインネームが確認できればDNSパケットであると考えられる．また，DNSパケットであった場合，DNSヘッダ内部のフラグにQR(Query / Response)の値がある．このフラグは0なら照会，1なら応答を示す．すなわち，UDPパケットのペイロードの文字列にドメインネームがあり，QRの値が1であれば，DNSサーバからの返信パケットだと考えられる．TCPパケットのラベリングの際に述べた通り，ダークネットに返信パケットが通常送信されることがないため，このパケットはIPアドレスを偽装してダークネットから送られた，DNSサーバへのDDoS攻撃によるバックスキャッタであると考えられる．ここで，DNSで主に使われるのは53番ポートであるため，UDP53番ポートから送信されたパケットの内，

・ペイロードの文字列にドメインネームがある
・QR(Query/Response)の値が1である

これらの条件を満たすパケットを，バックスキャッタであるとしてラベリングを行う．

2-4. 学習器

本研究では，最初にSVM(Support Vector Machine)を用いてバックスキャッタの判別を行う．SVMのカーネル関数にはRBF(Radial Basis Function)カーネルを使用した．

まず，今までの工程で作成した特徴ベクトルとラベル情報のペアを，教師データ用とテストデータ用に分割する．SVMのハイパーパラメータである，コストパラメータCとRBFカーネルパラメータγは，先行研究よりグリッドサーチと交差検定を用いてベストパラメータを決定する．グリッドサーチには$C = 2^{-5}, 2^{-3}, \cdots, 2^{15}$，$\gamma = 2^{-7}, 2^{-5}, \cdots, 2^{9}$を用いる．また，5-fold交差検証を用いる．

3. 実験概要

実験に用いるデータセットは，NICTから提供されたダークネットトラフィックデータを使用する．2019年のダークネットパケットから，TCP80番ポート，UDP53番ポートからのパケットを抽出する．それらのパケットデータを2-2で示した手順で処理を行い，5489個のデータを作成する．次に，2-3で示したルールに則ってバックスキャッタであるか，非バックスキャッタであるかのラベリングを行う．ラベリングを行ったデータの内，4116個を教師データ，1373個をテストデータとする．その後，2-4の手順のとおり，教師データを用いて学習を行っていく．特徴ベクトルそれぞれの学習に対する有効性を見るために，以下の106パターンで実験を行った．

・14個の特徴ベクトルを使った1パターン
・13個の特徴ベクトルを使った14パターン
・12個の特徴ベクトルを使った91パターン

予測結果の評価尺度として，以下の4つの指標を用いる．

(バックスキャッタを正，非バックスキャッタを負として考える．)

・適合率：正と予測したデータのうち，実際に正であるものの割合

・再現率：実際に正であるもののうち，正であると予測されたものの割合
・F尺度：再現率と適合率の調和平均

4. 実験結果及び概要

評価実験の結果より，14個の特徴ベクトルを使った1パターンの実験結果を表1に示す．どの特性も97%以上という結果を得ることができた．特に再現率は100%ということから，テストデータにあるすべてのバックスキャッタを検出することができた．

表 1　14 個の特徴ベクトルを使った 1 パターン

適合率	再現率	F 尺度
0.9774	1	0.9882

表 2　13 個の特徴ベクトルを使った 14 パターン

除いた特徴ベクトル	適合率	再現率	F 尺度
パケット間の時間差の分散	0.9767	1	0.9882
宛先 IP ごとのパケット数の平均	0.9767	1	0.9882
ペイロード長の分散	0.9767	1	0.9882
送信元ポートごとのパケット数	0.9767	1	0.9882
送信元ポートごとのパケット数の分散	0.9767	1	0.9882
ペイロード長の平均	0.9767	1	0.9882
宛先 IP ごとのパケット数の分散	0.9767	1	0.9882
宛先ポートごとのパケット数の分散	0.9767	1	0.9882
宛先ポートごとのパケット数	0.9767	1	0.9882
送信元ポートごとのパケット数の平均	0.9767	1	0.9882
宛先 IP ごとのパケット数	0.9767	1	0.9882
パケット数	0.9767	1	0.9882
パケット間の時間差の平均	0.9767	1	0.9882
宛先ポートごとのパケット数の平均	0.9692	1	0.9843

続いて，13個の特徴ベクトルを使った14パターンの実験結果を降順で並べたデータを表2に示す．表 2 より 14 パターン中，上から 13 パターンでは適合率が 0.001 弱低下し，再現率，F 尺度は変化なし，という結果が出た．しかし，一番下の「宛先ポートごとのパケット数の平均」を除いたパターンのみ，適合率が 0.0082 低下し，F 尺度が 0.0039 低下した．よって「宛先ポートごとのパケット数の平均」の特徴ベクトルが適合率等の精度の向上に寄与していることが推察される．

次に 12 個の特徴ベクトルを使った 91 パターンより，適合率を降順で並べた際の最大値と最小値周辺のデータを表 3 に示す．12 個の特徴ベクトルを使った 91 パターンでは，適合率を降順で並べた際の上から 80 パターンでは，適合率のみ 0.001 弱低下し，再現率，F 尺度に変化はなかった．しかし，三点リーダより下にあるデータを見ると，「パケット間の時間差の平均」と「宛先 IP アドレスごとのパケット数の平均」のペア，もしくは「宛先ポートごとのパケット数の平均」を含んだペアにおいて，適合率の 0.0082 の低下や，それまで 1 であった再現率の低下がみられた．これらの結果から，「宛先ポートごとのパケット数の平均」と「宛先 IP アドレスごとのパケット数の平均」が特性の性能に影響を与えていることがわかる．以上より，宛先の平均に関する特徴ベクトルが本システムの精度の向上において重要な要素であることがわかる．

次に，特徴ベクトルの組み合わせと学習時間の関係について考える．特徴ベクトルが 14 個の場合の学習時間と，13 個，12 個の場合の最長，最短の学習時間を表 4 に示す. 特徴ベクトル数が 13 個，12 個のときの最短時間のデータには，どちらにも先ほどの特性の低下に影響した特徴ベクトルである「宛先ポートごとのパケット数の平均」が含まれている．このことから，この特徴ベクトルは学習時間の増加と引き換えにバックスキャッタ検出精度の向上に寄与していると考えられ

表 3　12 個の特徴ベクトルを使った 91 パターン

除いた特徴ベクトル	適合率	再現率	F 尺度
パケット間の時間差の平均 パケット間の時間差の分散	0.9767	1	0.9882
・ ・ ・	・ ・ ・	・ ・ ・	・ ・ ・
パケット間の時間差の平均 宛先 IP ごとのパケット数の平均	0.9767	0.9921	0.9843
宛先ポートごとのパケット数の平均 パケット数	0.9767	0.9921	0.9843
宛先ポートごとのパケット数の平均 送信元ポートごとのパケット数の平均	0.9767	0.9921	0.9843
宛先ポートごとのパケット数の平均 宛先 IP ごとのパケット数	0.9767	0.9921	0.9843
宛先ポートごとのパケット数の平均 宛先ポートごとのパケット数	0.9692	1	0.9844
宛先ポートごとのパケット数の平均 パケット間の時間差の分散	0.9692	1	0.9844
宛先ポートごとのパケット数の平均 宛先 IP ごとのパケット数の平均	0.9692	1	0.9844
宛先ポートごとのパケット数の平均 ペイロード長の平均	0.9692	1	0.9844
宛先ポートごとのパケット数の平均 送信ポートごとのパケット数の分散	0.9692	1	0.9844
宛先ポートごとのパケット数の平均 送信元ポートごとのパケット数	0.9692	1	0.9844
宛先ポートごとのパケット数の平均 ペイロード長の分散	0.9692	1	0.9844

表 4　特徴ベクトルの組み合わせと学習時間

特徴ベクトル数	除いた特徴ベクトル	学習時間（秒）
14 個	—	4.383343
13 個	パケット間の時間差の分散	5.956301（最長）
	宛先ポートごとのパケット数の平均	3.491916（最短）
12 個	パケット間の時間差の平均 パケット間の時間差の分散	12.34378（最長）
	パケット間の時間差の平均 宛先ポートごとのパケット数の平均	3.097156（最短）

る．次に，最長時間のデータを見ると，14個全てを使ったときの学習時間より大幅に増加している．そのため，除いた特徴ベクトルに共通して含まれている「パケット間の時間差の分散」は，精度には影響をほとんど及ぼさない代わりに，学習時間の短縮に寄与していることがわかる．

5. まとめ

本研究では，DDoS攻撃を早期に検知するために，ダークネットで観測されたパケットから，DDoS攻撃による返信パケットであるバックスキャッタを判別するシステムを作成した．早期の検知のために，届いた最初のパケットから，短い時間内に届いたパケットを抽出し，特徴ベクトルを作成した．

評価実験には，バックスキャッタか非バックスキャッタかをラベリングすることができるTCP80番ポートとUDP53番ポートから届いたパケットを用いた．また，特徴ベクトルそれぞれの有効性を見るために，14個の特徴ベクトル全てを使った1パターン，13個の特徴ベクトルを使った14パターン，12個の特徴ベクトルを使った91パターンの計

106パターンでシステムのバックスキャッタ検出精度および学習時間を調べた.

その結果，テストデータ全体で97%を超える適合率，再現率，F尺度を得ることができた．これらの性能には，宛先ポートごとおよびIPアドレスごとのパケット数の平均の特徴ベクトルが大きく影響を及ぼすことを明らかにした．また，学習時間の短縮には，「パケット間の時間差の分散」の特徴ベクトルが大きく影響を与えることを明らかにした.

今後の課題として，判明した特性および学習時間に影響を及ぼす特徴ベクトルの組み合わせを利用し，高性能かつ短時間の処理を行うことができる最適な特徴ベクトルの組み合わせを見つけることが求められる．また，本研究では結果的にUDPに対するラベリングを行うことができなかったため，UDPに対するラベリングを今後の課題とする．さらパケットの前処理で扱った「最初から1分間」，「パケット数が10個以上」，「パケット間隔が1時間以上」などの数値を変更することによる特性や学習時間への影響を研究し，最適な値を見つけ効率化を図ることも今後の目標とする.

謝辞

本研究で使用したデータの提供にご協力いただいた MWS 組織委員会，システムの作成に当たりご協力及びご助言をいただいた情報通信研究機構（NICT）の笠間貴弘氏，ニッシンの畑太一氏に深謝します.

参考文献

1) Kaspersky サイバー脅威調査：2020 年第 1 四半期の DDoS 攻撃 (https://www.kaspersky.co.jp/about/press-releases/2020_vir22052020)
2) Kaspersky サイバー脅威調査：2020 年第 2 四半期の DDoS 攻撃 (https://www.kaspersky.co.jp/about/press-releases/2020_vir18092020)
3) Siti-Hajar-Aminah ALI et al., “Distributed Denial of Service (DDoS) Backscatter Detection System Using Resource Allocating Network with Data Selection,” Memoirs of the Graduate Schools of Engineering and System Informatics Kobe University, no.7, 2015.
4) 荒木粧子，他，“マルウェア対策のための研究用データセット～MWS Datasets 2019～,” 情報処理学会，vol.2019-CSEC-86, no.8, 2019.
5) Daisuke Inoue et al., “nicter: An Incident Analysis System Toward Binding Network Monitoring with Malware Analysis,” 2008 WOMBAT Workshop on Information Security Threats Data Collection and Sharing, pp.58-66, 2008.

2021年3月5日原稿受付，2021年3月6日採録決定
Received, March 5th, 2021; accepted, March 6th, 2021