リアルタイム性を考慮したエントロピーベースDDoS攻撃検知手法の提案

臼崎　翔太郎 [a)]・山場　久昭 [b)]・油田　健太郎 [c)]・岡崎　直宣 [d)]

A Proposal of Real-Time Entropy-based DDoS Attack Detection Method

Shotaro USUZAKI, Hisaaki YAMABA, Kentaro ABURADA, Naonobu OKAZAKI

Abstract

From the background that the risk of DDoS attacks is increasing year by year, it is important to detect attacks in real time and quickly shift to attack mitigation processing. Entropy methods with high detection accuracy and speed computation are widely used as one of the DDoS attacks detection approach to improve real-time performance. On the other hand, although the entropy method is recommended to use a wide window size to reduce the influence of noise, not only the processing efficiency degrades when the window size is increased but also the attack detection delays since the interval of detection processing increases. In addition, the attack detection using the entropy is important to learn the average and variance parameters from the latest data with a small degree of abnormality in order to determine the optimum threshold. Our method reduces the influence of noise with shortening the detection interval by applying the existing data mining method with efficient aggregation processing and sequentially learns the latest data in the period except for the burst traffic to automatically adjust the parameter. Experimental results show that our method has the Precision up to 0.978 and the Accuracy up to 0.992, these values are inferior to the maximum 1.0 with existing method, however, sufficient detection accuracy. On the other hand, in CICIDS 2017, the Precision was as low as 0.790 at the maximum. Although our method extracted the attack observation period, we need to calculate the abnormality for each host in the future. In terms of the processing time, proposal method is faster than the 296μsec of existing method. The performance is also higher than the basic entropy detection method. In future work, we need to evaluate the performance using real data traffic. .

Keywords: DDoS, Entropy, Aggregation pyramid, Real-time detection, Burst detection

1. はじめに

不正なトラフィックを送信しサーバをサービス停止状態に追い込む DDoS（Distributed Denial-of-Service）攻撃は、インターネットが社会インフラとなっている昨今ではその検知および緩和処理が重要となっている。実際に、2000 年には大手の検索サービスの Yahoo[1)] や、DNS のルートサーバ [2)] への攻撃が報告されている。また 2016 年には DNS サーバを提供する Dyn に対する攻撃が観測されており [3)]、この攻撃の影響により、Twitter や Spotify などが一時的に利用できなくなる事態となっている。このように、DDoS 攻撃の被害は年々大きなものとなっており、攻撃リスクの最小化が望まれている。

DDoS 攻撃の規模は、図 1 に示すように、IoT ボットネットによる攻撃やリフレクション攻撃の流行が原因で 2012 年を境に年々増加しており [4)]、2018 年現在、DDoS 攻撃として史上最大である 1.3Tbps 以上の規模の攻撃が観測されている [5)]。DDoS 攻撃を受けた 1,021 社を対象に行った Neustar 社のレポートによれば、2017 年には 79%の企業が、DDoS 攻撃によって 1 時間で約 2,500 米ドルの被害を受けたと報告されている [6)]。さらに、2017 年第 1 四半期に VERISIGN 社から提出された報告によると、観測時点では、DDoS 攻撃の最大規模は約 900,000 pps であり、121 Gbps が観測されている [7)]。さらにクラウドにおける DDoS 攻撃は、攻撃自体の被害に加え、リソースを利用した分だけ料金を支払う従量課金制であることに起因する EDoS 攻撃被害も引き起こす。EDoS（Economic Denial-of-Sustainability）攻撃はクラウド利用組織に向けて経時的に大量のリクエストを送信し、リソースの追加を長時間にわたり強要させる攻撃である。これによって間接的にクラウド利用組織の課金額を増やして経済的損失を発生させる。2014 年には Amazon EC2 を利用した組織への DDoS 攻撃で 1 日あたり 3 万ドルの EDoS 攻撃被害が発生していることが報告されている [8)]。DDoS 攻撃被害を緩和するためにはリアルタイムに攻撃を検知し、攻撃緩和処理への素早い移行をサポートすることが重要である。リアルタイムに攻撃を検出する方法として、比較的計算量の少ない、統計的検知手法が広く利用されている。統計的検知手法では、時間やパケット数を単位とする系列（以降ウィンドウサイズと呼ぶ）ごとに特徴量を計算し、過去データとの乖離度を利用して攻撃検知を行う。その中でもエントロピーを利用して、出現するホストの偏り具合を監視するエントロピー手法が広く利用されている。エントロピー手法では基本的な処理がパケット

[a)] 工学専攻機械・情報系コース大学院生
[b)] 情報システム工学科助教
[c)] 情報システム工学科准教授
[d)] 情報システム工学科教授

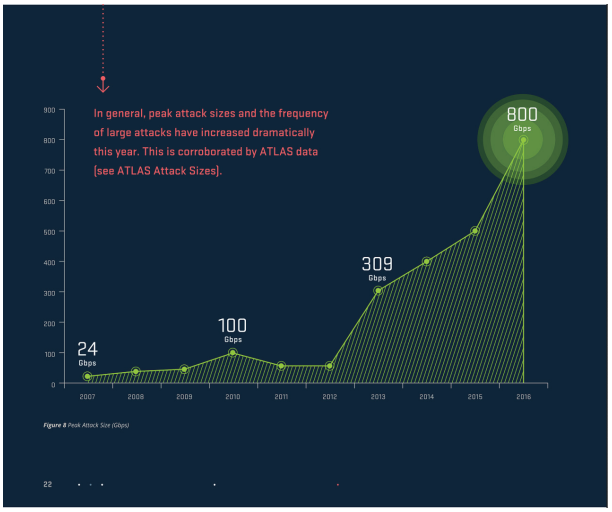


図 1. Arbor 社による DDoS 攻撃のレポート（文献 [4] より引用）

カウントであることから高速計算性が高く、さらに攻撃検知精度の高さが特徴となっている。

一方でエントロピーベース手法は、外れ値の影響を小さくするために、ウィンドウサイズが時間単位の場合には 1 分程度 [9]、パケット数単位であれば数万程度 [10] のウィンドウサイズが推奨されているが、ウィンドウサイズを大きくすると処理効率が悪くなることが指摘されている [11]。リアルタイム性を向上させるために、一般的にはウィンドウサイズを小さくして検知処理の頻度を多くすること考えられるが、その代わりにノイズの影響が大きくなり検知精度に悪影響を及ぼしてしまう。

また、エントロピーベース手法において閾値決定のために平均・分散などのパラメータを学習することが重要となるが、インターネットトラフィックの通常状態を定義することが困難であるとともに、エントロピー値は時間帯によっても変化するため、追従性のためには、できるだけ異常度の少ない直近のエントロピー値を用いてパラメータを計算する必要がある。

そこで本論文では既存のテキストデータマイニング手法 [12] を応用して効率の良い集約処理を行うことにより、ノイズの影響を軽減しながら小さいウィンドウサイズの検知処理を可能にし、エントロピー手法のリアルタイム性と効率性を向上させる。さらに、バースト的特徴を検知するという長所を利用し、バースト的通信の観測されていない直近のエントロピー値を用いて、攻撃検知に利用する平均および分散値のパラメータを学習する。これによって、トラフィックの追従性を向上させる。

2. 本論文の構成

本論文は以下の構成となる。第 2 章では DDoS 攻撃について説明する。第 3 章では DDoS 攻撃の既存の検知手法について述べるとともに問題点を分析する。第 4 章では、提案手法について詳細に述べ、第 5 章では提案手法の性能を調査するための実験環境と実験方法について説明する。第 6 章では、提案手法の性能を実験結果について述べたあと、その結果について考察し、第 7 章で結論を述べる。

3. DoS/DDoS 攻撃

DoS (Denial-of-Service) 攻撃とは、攻撃対象となるサーバに不正なトラフィックを送り込み、サービス停止状態 (DoS 状態) に強制的に陥れる攻撃である。DDoS (Distributed Denial-of-Service) 攻撃は、DoS 攻撃を複数台のホストから行う攻撃である。攻撃者は脆弱なマシンを Telnet や SSH を利用して乗っ取り、攻撃の踏み台として操作する（こうしたホストはボット、ゾンビと呼ばれる)。この多数のボットで形成されたネットワークをボットネットと呼ぶ。ボットネットは C&C と呼ばれるサーバ/クライアント型のネットワーク形態にあたり、攻撃の中枢とも呼べる C&C サーバから攻撃司令命令を受け取ると、ボットである C&C クライアントが一斉に攻撃トラフィックを送信する。攻撃トラフィックの送信には、DoS 攻撃ツールが用いられる。DDoS 攻撃の特徴的な点は 1 つ 1 つのホストからのパケット量は正規のユーザと区別が困難であるという点である。また、攻撃に加担しているホストは悪用されたり乗っ取られたりしたものであるため、攻撃者本人を辿ることが困難となっている。

寺田によれば、DoS/DDoS 攻撃は Exploit 型と Flooding 型で大別される [13]。Exploit 型はサーバで稼働しているプロセスの脆弱性を突いてサービスの停止を狙う攻撃である。一方、Flooding 型はサーバの許容量や通信の帯域を超える大量のトラフィックを送り込んでサーバを停止状態に陥らせる。DDoS 攻撃は Flooding 型の攻撃が割合が高いとされている。実際に 2018 年現在、SYN Flood 攻撃や HTTP Flood 攻撃などの Flooding 型攻撃が多数観測されている [14]。この理由としては、Exploit 型はサーバプログラムの脆弱性を探し出してそれをクラックする技術が攻撃者に必要となるが、Flooding 型は単純にパケットを大量に送るだけで攻撃の効果が出るため、攻撃者にとってより容易に実行できることが挙げられる。実際、攻撃トラフィックを発生するツールや、ボットネットを形成する DDoS 攻撃のツールは多数公開されており、誰でも攻撃を行うことができる。最近では、ダークウェブ上で展開されている DDoS 攻撃代行サービスが利用されることも多い [15]。この場合、攻撃者はサービス利用料を支払うことで、攻撃環境を用意することなく攻撃を行うことができる。

DoS 攻撃の歴史は古く、インターネットの世界で初めて DoS という用語が使われたのは、1975 年に発表された IETF(Internet Engineering Task Force) 文書であるとされる [16]。DoS という言葉が急激に普及したのは、1996 年に発行された注意喚起文書である。1994 年に IP パケットの送信元 IP アドレスを詐称すること（以降 IP スプーフィングと呼ぶ）が可能である事実が発覚してから、本格的な DoS 攻撃が観測されるようになった。

最近の潮流として、不正トラフィックを増幅するリフレクション攻撃や、Mirai に代表される IoT 機器によるボットネットからの攻撃が観測されている。2016 年に個人ブログに当てて行われた攻撃は IoT (Internet of Things) 機器を狙ってボットネットを形成するマルウェアである Mirai が用いられており [17]、2018 年に GitHub 社に行われたリフレクション攻撃は memcached という分散型メモリキャッシュシステムのレスポンスを悪用して攻撃トラフィックを増幅させている [5]。

3.1 DDoS 攻撃の種類

DDoS 攻撃は 1996 年に観測されて以来、複数種類の攻撃が観測されている。その中でもここでは Flooding 型の代表的な攻撃について紹介し、さらに提案手法で対象とする攻撃を示す。

3.1.1 TCP SYN Flood 攻撃

TCP（Transmission Control Protocol）通信では、3 ウェイハンドシェイクによってホスト間の通信が確立する。ここでは、通信先をサーバ、通信元をクライアントとする。まず、クライアントはサーバに対して SYN パケットを送信する。これは、通信を行いたいという意思表示とも言える。SYN パケットを受け取ったサーバはクライアントに対して、SYN/ACK パケットを返す。これによってサーバは自身が通信を行う準備があることをクライアントに伝える。SYN/ACK パケットを受け取ったクライアントはその旨を了承したことを ACK パケットを返送することで伝える。

TCP SYN Flood 攻撃は 3 ウェイハンドシェイク時に SYN パケットのみを大量に送りつける攻撃である。攻撃者はサーバから返ってくる SYN/ACK パケットには返答せずそのまま放置する。TCP の設計思想から、サーバは通信に不具合が発生したのだと考えて SYN/ACK パケットを再送し通信が正常に到着するのを待つ。ACK パケットは数十秒待つように設計されており、その間クライアントの情報を保持し続ける必要があることからメモリ利用量が増大する。このことから、SYN パケットが極めて短時間に大量に送信されると、サーバのメモリ空間が食いつぶされることとなり、DoS 状態に陥る。

この攻撃は 1996 年に初めて大規模に攻撃が行われて以降、2018 年現在でも広く利用されている方法である。SYN パケットに返答しなくて良い点から容易に IP スプーフィングを行うことが出来る。

3.1.2 UDP Flood 攻撃

ターゲットの UDP ポートに対してデータサイズの大きいパケットを大量に送り続ける攻撃である。UDP というプロトコルがコネクションレスであることから IP スプーフィングは容易に可能となっており、攻撃者を特定することが難しい。対策として、不要な UDP ポートを閉じておくことが挙げられるが、DNS などの広く利用される UDP ポートでも攻撃が観測されることから、根本的な対策とは言えない。

3.1.3 ICMP Flood 攻撃

ターゲットに対して ICMP echo リクエストパケットを大量に送信する攻撃である。ICMP も UDP と同様にコネクションレスの通信形式であるため、IP スプーフィングが容易である。対策として Ping パケットを組織内に通さないように ICMP パケットを遮断することが挙げられる。

3.1.4 DNS リフレクション攻撃

DNS の仕様を悪用して、DNS サーバを攻撃パケットの踏み台とした DoS/DDoS 攻撃を仕掛ける攻撃である。まず攻撃者は送信元 IP アドレスを攻撃対象のものに詐称したクエリを DNS サーバに送信する。この DNS サーバは到達した DNS リクエストパケットに対して返答することになるが、結果的に攻撃対象とする IP アドレスに対してパケットを送信することとなり、攻撃に加担することとなる。これを複数の DNS サーバに対して行えば DDoS 攻撃となる。この種の攻撃は Amplification 攻撃、あるいは DRDoS(Distributed Reflection DoS) 攻撃とも呼ばれ、DNS だけでなく IP スプーフィングが容易な UDP を利用したプロトコルでは同様の攻撃が行われやすい。特に、DNS や NTP、TFTP などのようにリクエストに対してレスポンスの方がパケットサイズが大きくなる特徴を持つプロトコルは、踏み台として攻撃者に悪用されやすく、この増幅率を Amplification Factor と呼ぶ。US-Cert によれば Amplification Factor は DNS の場合は 28 から 54、NTP は 556.9、また 2018 年に観測された史上最大規模の DDoS 攻撃で悪用された memcached は 10,000 から 51,000 に及んでいる [18]。

3.1.5 Smurf 攻撃

Smurf 攻撃は ICMP を利用したリフレクション攻撃の一種である。送信元 IP アドレスを攻撃対象のものに詐称した ICMP エコーリクエストパケットを、増幅用ネットワークのブロードキャストアドレス宛てに送信する。この時、増幅用ネットワークに所属する全てのホストが、攻撃者によって詐称された送信元 IP アドレスに対して、ICMP エコーレスポンスパケットを送信し攻撃に加担することとなる。攻撃の対策として、ICMP Flood 攻撃と同様に、ICMP パケットを遮断することが考えられる。

3.1.6 HTTP GET Flood 攻撃

HTTP における GET リクエストを大量に送り付け、HTTP サーバの処理負荷を増やし、リソースを消費させる攻撃である。この攻撃はまずコネクションを確立しなければならないため、IP スプーフィングは困難であり、攻撃パケットの発生元は特定しやすく検知後の対処は効果的だが、一般的に通常パケットと攻撃パケットの区別が困難であるといわれている [19]。

3.1.7 TCP Connection Flood 攻撃

ターゲットの TCP ポートに対して大量にコネクションを張りプロセスを起動することで、サーバのリソースを消費させる攻撃である [20]。この攻撃も HTTP Get Flood 攻撃と同様にコネクションを確立しなければならないため、IP スプーフィングは不可能に近い。対策としては、サーバの同時接続数を制限することなどが挙げられる。

3.1.8 EDoS 攻撃

EDoS（Economic Denial-of-Sustainability）攻撃は、クラウドサービスが従量課金制である点を悪用する攻撃である。この攻撃の特徴的な点はサービスの停止を目的とした攻撃ではなく、金銭を大量に消費させて持続可能性を減少させることが狙いであることである。クラウド環境において Flooding 型の DDoS 攻撃を行った際の副作用として発生することが多く、被害の大きい事例の多くは DDoS 攻撃が原因となっている。EDoS 攻撃では図 2 に示すように 3 人の関係者がいる；攻撃者、クラウドサービスを利用するクラウド利用者、クラウドサービスを提供するクラウド事業者である。まず、攻撃者が大量にトラフィックを送信すると、クラウド利用組織ではトラフィックに対応できるようにリソースの拡充を行う。この

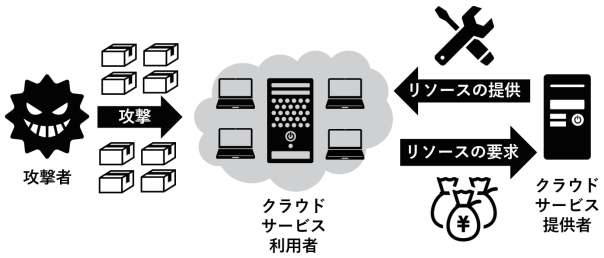


図 2. EDoS 攻撃の概要図

際、その分の対価としてクラウド事業者に課金を行う必要がある。攻撃者はこれを経時的に行い、クラウド利用者に金銭の負担を強いる。

この攻撃被害で、2014 年には AWS (Amazon Web Services) で提供される EC2 インスタンスを利用した組織が 1 時間あたり 200 万円の損害を出した [8]。

3.2 提案手法で対象とする攻撃

先述の代表的な攻撃を以下に分類する。

- サーバのシステムリソースを消費
  - TCP SYN Flood 攻撃
  - HTTP Get Flood 攻撃
  - TCP Connection Flood 攻撃
- ネットワークの帯域を消費
  - UDP Flood 攻撃
  - ICMP Flood 攻撃
  - DNS Amplification 攻撃

提案手法では上述の分類のうち、特にネットワークの帯域を消費する攻撃を対象とする。その理由としては、パケット量・データ量ともに大規模になることが多く検知が遅れればサーバ以外のシステムにも影響を及ぼすこと、特にリフレクション攻撃の場合は外部組織のシステムの管理不足が原因であることが多く、サーバのリソースを消費させる攻撃に比べて能動的な対策が難しいためである。

4. 関連研究

本章では既存の DDoS 攻撃検知手法について述べる。まず DDoS 攻撃検知手法は、シグネチャ型検知手法とアノマリ型検知手法の二つのタイプに分類される。シグネチャ型手法 [21] はあらかじめ既知の攻撃の特徴をパターンとして保持しておき、そのパターンと比較してマッチした時に攻撃として検知する手法である (図 3)。広く利用されているものに Snort[22] がある。この方法では事前に正しいパターンが登録されているときには非常に高い攻撃検知精度を持つ手法である。この登録されたパターンのことをシグネチャと呼ぶ。しかしながらパケットごとに処理を行う必要があるため、パターンの登録数やパケットが大量に到着した際にはリアルタイム性に欠けたり、機器に負担がかかったりするおそれがあることが指摘されている [23][24]。また、事前にシグネチャを登録しておく必要があるという特徴から未知の攻撃には対応することが

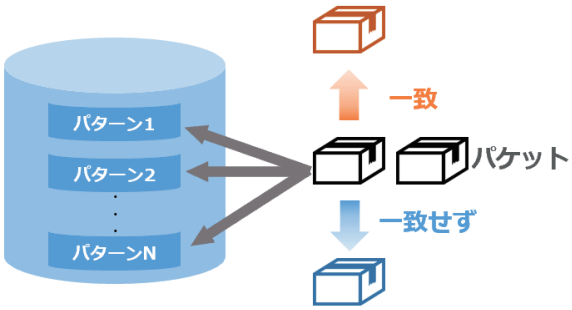


図 3. 一般的なシグネチャ型検知手法の概要図

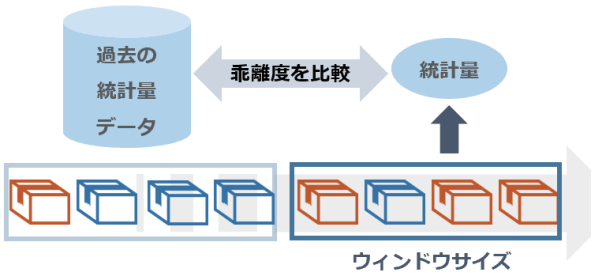


図 4. 一般的なアノマリ型検知手法の概要図

不可能であるという点、常に最新のパターンを反映しなければならない点が欠点となる [25]。

アノマリ型検知手法は特徴量を用いて異常を検知する手法で、ネットワークトラフィックの正常状態をモデル化し、そこから乖離した状態を異常とみなす方法である (図 4)。シグネチャ型検知手法と比較した時のアノマリ型検知手法の利点は、未知の異常を検知できる可能性がある点や拡張が容易である点があげられる。また、変化点検知手法や統計的検知手法に関しては処理効率が高く、リアルタイム性が高い。しかしながら、インターネットトラフィックにおいて正常状態を定義することが難しい点や [26]、パラメータを時々刻々に調整しなければならない点が欠点として挙げられる。

本論文では、リアルタイム性の高さからアノマリ型検知手法に着目し、特に特徴が顕著なボリュームベース検知手法、エントロピーベース検知手法を紹介する。

4.1 ボリュームベース検知手法

DDoS 攻撃検知の最も単純な形式として、ボリュームベース検知手法がある。これは、一定期間ごとにパケット量を監視し、あらかじめネットワーク管理者が定めた閾値よりも多かった場合には攻撃として検知する手法である。これは DDoS 攻撃のみならず DoS 攻撃を防ぐ単純な機構として利用される。この手法は処理効率が非常に高く、それでいて効果が認められることから広く用いられるが、トラフィック量が経時的に異なることから閾値を決めにくい点や、攻撃者に閾値を知られる可能性が比較的高く、これを踏まえて閾値を上回らない程度の DDoS 攻撃が行われると対処ができない点、通常通信を攻撃と判定してしまう割合 False Positive Rate が大きい点が欠点となる。

4.2 エントロピーベース検知手法

DDoS 攻撃検知手法で用いられる特徴量の中で、高速計算性を持つ方法ながら高い精度を持ち、広く利用されているのがエントロピーベース手法である。エントロピーベース手法は、パケットのヘッダ情報を情報源としてエントロピー値を計算し、それらの増減を監視して攻撃を検知する手法である。利用されるエントロピー値としては、最も有名なシャノンによる Shanon's Entropy が多く、DDoS 攻撃検知の観点からは Shanon's Entropy を拡張した Renyi's Entropy も利用されることがある。

エントロピーベース手法の一般的な攻撃検知の仕組みを説明する。エントロピー値には乱雑度が高いほど大きくなり、低いほど小さくなるという特徴があり、これを利用して DDoS 攻撃時に発生する特徴を捉えている。例えば、送信元 IP アドレスを情報源とするエントロピー値を H_s、宛先 IP アドレスを情報源とするエントロピー値を H_d とする。DDoS 攻撃時には、大量のホストから攻撃が到来するため大量の送信元 IP アドレスがサンプル内に出現し H_s が増加していく。また、ある特定の宛先に攻撃を行うことから、H_d ではある特定の宛先 IP アドレスの出現回数が極端に多くなるため、乱雑度が減ったとして H_d が減少していく。このように、複数のエントロピー値を組み合わせて攻撃の検知を行う。

ここで各エントロピー値の具体的な算出方法を説明する。S を計算対象となるサンプルデータの系列、n_i をシンボル i の出現数とすると、時間 t における Shanon's エントロピー値算出式を以下で表すことができる。

$$H(t) = -\sum_{i=1}^{m} p_i \log p_i = -\sum_{i}^{m} \frac{n_i}{S} \log \frac{n_i}{S} \tag{1}$$

上式にパラメータ値 α を加えエントロピー値の調整を可能にしたものが Rényi Entropy である。

$$H_\alpha(t) = \frac{1}{1-\alpha} \log \left(\sum_{i=l}^{n} p_i^\alpha \right) \tag{2}$$

また、エントロピーベースの手法として、Information Distance という確率分布間の距離を測る値も DDoS 攻撃検知に広く利用されている。Infotmation Distance でよく用いられるカルバックライブラーダイバージェンスは、(3) 式において α=1 に設定したときである (4) 式で示される。

$$D_\alpha(P||Q) = \frac{1}{\alpha - 1} \left(\sum_{i=l}^{n} p_i^\alpha q_i^{1-\alpha} \right) \tag{3}$$

$$D_0(P||Q) = \sum_{i=0}^{n} p(i) \log \frac{p(i)}{q(i)} \tag{4}$$

一般的にエントロピー手法では、送信元 IP アドレス、宛先 IP アドレス、送信元ポート番号、宛先ポート番号、プロトコルといった 5-Tuple の要素が情報源として利用される。この中でも最も利用されるのが送信元 IP アドレスと宛先 IP アドレスである。小島ら [11] は上記の項目に加えてパケット数、TTL など計 9 つの情報量を利用しているが、最も攻撃検知精度に寄与した成分が前述の 2 つの情報源であったと結論づけている。そのほか、SYN Flooding 攻撃など送信データが 40byte とあらかじめ決まっている攻撃には、パケットのサイズも攻撃検知精度の重要なファクターとなる。

エントロピー手法の異常検知性能については Geroge[9] や Domenico ら [27]、Monowar ら [28] に詳しい。

Geroge らは、エントロピー手法についてどの情報源を利用すればどのような異常を検知可能なのか、実際のトラフィックおよび仮想的に作成した攻撃トラフィックを用いて検証している。検証に用いた情報源は複数の文献で利用されている「送信元および宛先 IP アドレス」、「送信元および宛先ポート番号」、「フローごとのフローサイズ)」といったパケットヘッダに基づく情報源と、「Indegree/Outdegree」という振舞に基づいた情報源の二つの利用している。ここでフローとは、5tuple で区別される ID であり、フローサイズはそのフローの出現回数である。また、In-degree とはあるホスト x にコンタクトしてきたホストを、Out-degree はあるホスト x がコンタクトしたホストを意味する。なお検知処理に利用する特徴量は生のエントロピー値ではなく、上述の計七つのエントロピー値についてそれぞれウェーブレット変換を適用した結果を利用し、ノイズの影響を抑えている。この調査の結果、エントロピー手法には次に示す異常検知の特徴が認められた:

- IP アドレスとポート番号の分布は一方向 (すなわちインバウンドかアウトバウンドか) のフローを監視した時に限り、通常のトラフィックにおいて高い相関 (> 0.95) が認められる。
- IP アドレスとポート番号によるエントロピー値によって検知される異常は重なっているため、4 つのうちの 1 つを採用すると良い。またフローサイズや Outdegree を利用した場合には、前述のパケットヘッダに基づく情報源によるエントロピー値では検知の難しい DoS 攻撃や P2P の異常な活動を検知することができる。
- IP アドレスやポート番号によるエントロピー値は、パケット量が急激に増加するような攻撃に対して有効である。

Domenico ら [27] は、背景からイタリア国内の Tier2 に所属する AS (Autonomous System) で観測された現実のトラフィックデータを対象に、エントロピー手法の検知性能を調査している。この AS ネットワークはイタリアの主要な三つのネットワークに接続しており、実環境に近いトラフィックが観測される。2010 年 9 月から 2011 年 8 月までのトラフィックを利用しており、その期間には複数の実際の DDoS 攻撃が観測されている。エントロピー値の候補として、Shanon Entropy、Rényi Entropy、カルバックライブラーダイバージェンスの三種類のエントロピー値を計算しており、ウィンドウサイズはエントロピー値の安定する 1 分としている。結果として、Rényi Entropy よりも Shanon Entropy の方がより安定した値となったと報告している。Domenico らは通常時に安定し、異常時に反応を見せたという点で、カルバックライブラーダイバージェンスが Shanon Entropy や Rényi Entropy よりも検知性能が高いと結論づけた。

Monowar ら [28] はエントロピー値による検知が低レートの DDoS 攻撃に対して有効かどうかを調査している。実験では、低レート DDoS 攻撃の含まれるデータセット CAIDA2007

と、TUIDS を利用し、高レートの DDoS 攻撃と、低レートの DDoS 攻撃で攻撃検知の性能を調査している。実験に用いるデータは先述の (2) 式を α を 0 から 15 に変化させたものと、(3) 式を α を 1 から 14 と変化させたものを利用している。結果的に、α を大きくするほど高レート攻撃と低レート攻撃を区別できる可能性が示唆された。

また、エントロピー手法はボリュームベース検知手法に比べて閾値を下回るように DDoS 攻撃を行うことが難しい。例えば İlker らは、エントロピーベース手法を回避する DDoS 攻撃を提案しているが [29]、攻撃の前に標的のネットワークを監視してエントロピー値の平均、標準偏差、分散を算出する必要がある。さらに攻撃を開始した後にもエントロピー値を随時監視し、もしも予想される範囲よりも現在のエントロピー値が低いならば、ランダムな IP アドレスを用いてパケットを送信し、逆に高いならば、特定の IP アドレスからパケットを送信するようにする。この方法によって検知を回避しているが、標的のネットワークに到着するパケットをポートミラーリングする必要があり、実現可能性は低いと考えられる。なお、この DDoS 攻撃はエントロピー値調整のためにパケットを送信する必要があるため、パケットレートの情報を利用することで検知が可能であると結論付けられている。

一方、エントロピー手法に関しては、ウィンドウサイズを大きくするとオーバヘッドが発生する点が指摘されている [11][30]。エントロピー値のウィンドウサイズの目安として、ウィンドウサイズの単位が時間の場合には 1 分程度 [9]、パケット数が単位の場合には数万 [10] のデータサイズが推奨されている。また、エントロピー手法は広く利用されているが検知フローが判然としておらず、最適な閾値の決め方が難しい。

まとめるとエントロピー手法には以下の特徴がある。

1. 単純な計算であるため高速計算性が高い。
2. 通常時のトラフィックの分布を知ることが攻撃者にとって難しいため、閾値を下回るように攻撃を仕掛けるのが困難。
3. 最適な閾値の決定方法が困難。
4. Flooding 型の DDoS 攻撃に対しては IP アドレスやポート番号、スキャン攻撃等に対してはフローサイズや Indegree や Outdegree を利用するなど、検知したい異常に合わせて利用するエントロピー値を吟味する必要がある。
5. ウィンドウサイズを大きくするとオーバヘッドが発生してしまう。

ここからはエントロピー手法の既存研究を説明する。

No らは、エントロピー値の圧縮処理を考案してエントロピーベース手法のリアルタイム性を改善している [31]。式は以下で与えられる。

$$H^{'} = -\log\frac{m}{n} + H^{''} \tag{5}$$

$$H^{''} = \begin{cases} |\log\frac{n_{i-1}}{n_i}| & (n_i \geq n_{i-1}) \\ |\log\frac{n_i}{n_{i-1}}| & (n_i < n_{i-1}) \end{cases} \tag{6}$$

$H^{''}$ の式を見れば明らかなようにパケットレートを考慮に入れて攻撃検知を行える特徴量であるが、トラフィックの様態によっては False Positive が高くなることが指摘されている [30]。

小島らは、エントロピー値を多次元マハラノビス距離に変換し、モデルとの距離を求めることによって DDoS 攻撃の検知を行っている [11]。マハラノビス距離とはデータの分散を考慮しながら計算される距離である。小島らは、9 次元のエントロピー値を利用してマハラノビス距離を算出することによって疑似的な標本数を増やし、ウィンドウサイズを小さくしても十分なデータを取得できる工夫をしてリアルタイム性を高めている。ここでウィンドウサイズの単位はパケット量となっている。しかしながら、9 次元マハラノビス距離を短いウィンドウサイズで計算するために処理効率に欠点があり、パケット量が多くなった時に対応が遅れる可能性がある。

Hoque らは、リアルタイム性の向上のため FPGA 実装による DDoS 攻撃検知手法を提案している [30]。攻撃検知に利用する特徴量をパケットレート、送信元 IP アドレスの変化回数、送信元 IP アドレスを情報源としたエントロピー値の DDoS 攻撃時に顕著な変化が出る 3 つに絞っており、さらにアルゴリズムを工夫して計算効率を高めている。

$$NaHid(X,Y) = 1 - \frac{1}{n}\sum_{i=1}^{n} D(i) \tag{7}$$

D(i) は X と Y のダイバージェンスであり、$\boldsymbol{X}$ の平均・分散を meanX、SDX、$\boldsymbol{Y}$ の平均・分散をそれぞれ meanY、SDY としたときに以下の式で表される。

$$D(i) = \frac{|X(i) - Y(i)|}{||meanX - SDX| - X(i)| + ||meanY - SDY| - Y(i)|} \tag{8}$$

FPGA 実装と処理効率の工夫によって、1 パケットあたり $354n$ 秒、ソフトウェア実装の場合には 296μ 秒の処理効率で攻撃検知を行うことができる。また、閾値以下となるデータをサーバに送信して通常データとして学習し、自動的に攻撃の閾値を調整している。この手法では DDoS 攻撃に特有な特徴のみを利用していることから、DoS 攻撃やポートスキャンなどの異常に対して反応できない可能性がある。

5. 提案手法

第 2 章から第 3 章までの議論を踏まえて、提案手法の位置づけを、ネットワークの帯域を溢れさせる目的の Flooding 攻撃を対象としたアノマリ型検知手法とする。このタイプの攻撃は短時間に大量のトラフィックが観測されるため、すばやい攻撃緩和を行う必要がある。その中でも異常なトラフィックをリアルタイムにとらえて、すばやく攻撃緩和処理に移行するための手法を提案する。

リアルタイム性を担保するためには、攻撃検知精度をパケットレートやエントロピー、フローサイズなどの比較的高速計算性の高い特徴量を利用して攻撃検知することが考えられる。しかしながらこの時次の 2 点が問題となる。

1. ウィンドウサイズを小さくしてリアルタイム性の向上を図った際に、ノイズの影響が大きくなってしまい、精度良く検知することができない可能性が考えられる。
2. エントロピー値やフローサイズを利用した検知では、平均・分散などのパラメータを学習することが重要とな

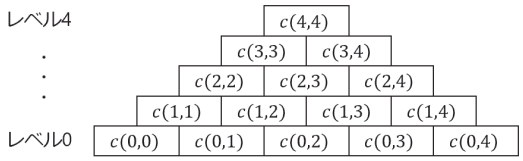


図 5. $L = 5$ の時の Aggregation Pyramid

る。モデルデータとしてできるだけ異常度の少ないエントロピー値を学習する必要があるが、トラフィックの通常状態を定義するのは困難であるとともに、時間帯によってもエントロピー値が変化するため、追従性のためにも、異常検知に利用するパラメータの自動調整が必要となる。

本研究ではテキストデータマイニングの手法の効率の良い集約処理により、ノイズの影響を軽減しながら小さいウィンドウサイズで検知処理を行うようにして 1. の解決を試みる。さらに、バースト的特徴を検知するという長所を利用し、通常状態と思われるトラフィックのみの状態を抽出して異常度のできるだけ少ない、直近のエントロピー値を学習データとして用い、攻撃検知に利用する平均および分散値のパラメータを学習するようにして 2. の解決を試みる。

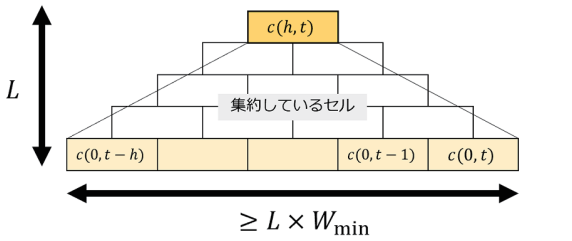
5.1 リアルタイムバースト検出手法 [12)]

提案手法について述べる前に、提案手法で利用するテキストデータマイニング手法を説明する。バースト検出手法はデータマイニングの分野で利用される技術であり、データストリームの異常状態を解析することが主な目的である。データストリームはオンラインニュースやブログ、電子掲示板といった高速に流れるデータのことを指す。データストリームにおけるイベントの集中発生状態のことをバーストと呼ぶ。このバーストをいちはやく検出することは流れの早い膨大なデータの中から現在注目されている事柄を抽出することに繋がる。

本研究で利用する蝦名らの手法 [12)]（以降、RTB 手法と呼ぶ）はリアルタイム性の高いバースト検出手法で、DDoS 攻撃検知の観点からは次の利点がある。まず、ウィンドウサイズを小さくしたまま広いウィンドウサイズでの統計量を効率よく参照することができる点が挙げられる。パケットレートやエントロピー値はウィンドウサイズを小さくするとノイズの影響が大きくなってしまうため、ウィンドウサイズを大きくする必要があるが、リアルタイム性が減少してしまうため、検知精度とリアルタイム性を担保できる可能性がある。また、バースト通信を抽出することができるという点が、エントロピー値の平均・分散を求めるのに良いと考えられる。エントロピー手法では送信元 IP アドレスや送信元ポート番号が算出されるが、

5.1.1 データ構造

RTB 法のデータ構造は Zhang らの手法で提案された Aggregation Pyramid[32)] というデータ構造を参考にしている。Aggregation Pyramid は、図 5 に示すように複数のセルで構成されており、L の階層を持っている。レベル h には $L-h$ 個のセルが存在しており、生成されたセルは各階層の右側に



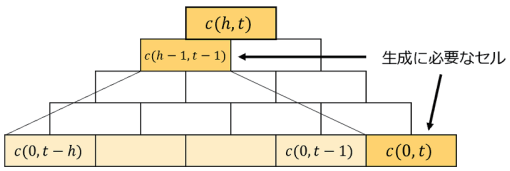


図 6. セルデータの集約処理

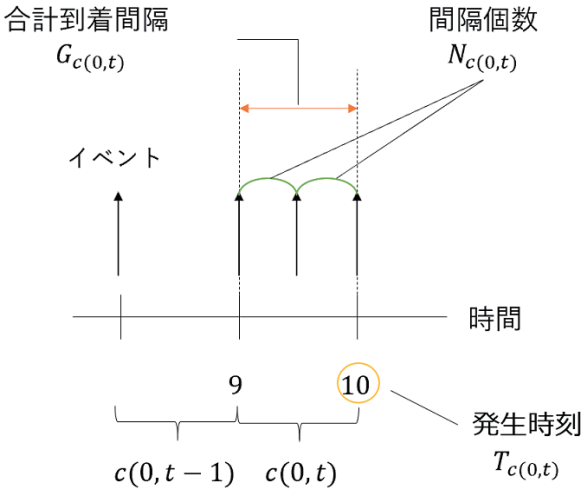


図 7. セルが保持するデータ

追加されていく。ここでセルの終了時間を t とすると、レベル h のセルは $c(h,t)$ と表現され、各セルはイベントの発生数、開始時刻、終了時刻、期間の長さの 4 つのイベント情報を保持する。イベントが発生すると、イベント情報を保持するレベル 0 のセル $c(0,t)$ を生成する。次に $c(0,t-h)$ から $c(0,t)$ までのイベント情報を保持するレベル h のセル $c(h,t)$ を生成する。$c(h,t)$ は、$c(h-1,t-1)$ と $c(0,t)$ のイベント情報を集約することで生成される（図 6）。

RTB 手法において、図 7 に示すように、セルには合計到着間隔 $G_{c(h,t)}$、到着時刻 $T_{c(h,t)}$、間隔個数 $N_{c(h,t)}$ を保持する。提案手法におけるデータ構造の構築過程を説明する。いま $n+1$ 個の一連のイベントが発生したときの間隔の時間を $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ と表現すると、以下のルールに従ってセルを生成し、データ構造を構築していく。ここで、W_{min} はセルが保持する最小のウィンドウサイズである。

1. レベル 0 セルの生成方法

   (a) $x_i \geq W_{min}$ の場合

   - $G_{c(0,t)} = x_i$
   - $T_{c(0,t)} = i+1$ 番目のイベント発生時刻

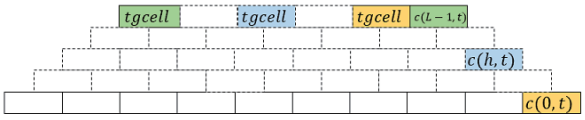


図 8. 各セルに対応する tgcell

- $N_{c(0,t)} = 1$
- $i = i + 1$

(b) $x_i < W_{min}$ の場合

- $T_{c(0,t)} = T_{c(0,t-1)} + W_{min}$
- $N_{c(0,t)} = T_{c(0,t-1)}$ から $T_{c(0,t)}$ 直前までの期間に発生したイベント発生数
- もし $T_{c(0,t)}$ にイベントが発生していないなら、$N_{c(h,t)} = N_{c(h,t)} - 1$
- $G_{c(0,t)} = W_{min}$
- $i = i + N_{c(0,t)}$
- $x_i = T_{c(0,t)}$ から次のイベントが発生するまでの経過時間
- もし $c(0,t)$ で複数のイベントが発生しているなら、$x_i = 0$

2. レベル h セルの生成方法

- $G_{c(h,t)} = G_{c(h-1,t-1)} + G_{c(0,t)}$
- $T_{c(h,t)} = T_{c(0,t)}$
- $N_{c(h,t)} = N_{c(h-1,t-1)} + N_{c(0,t)}$
- 最上位セル $(h = L-1)$ のとき、$t = t + 1$

5.1.2 バースト判定処理

バースト判定処理ではセル $c(h,t)$ が生成されると、集約している期間が重複しない直近の最上位レベル $(L-1)$ のセル $c(L-1, t-1-h)$ との間でそれぞれのイベント平均到着間隔を比較する。以降このセルのことを *tgcell* と呼ぶ。図 8 は各セルに対応する *tgcell* を示したもので、同じ色で示されているセルが、対応する *tgcell* となる。イベント平均到着間隔とはセルで集約しているイベントの到着間隔の平均値であり、セル $c(h,t)$ のイベント平均到着間隔は、セル $c(h,t)$ の合計到着間隔 $G_{c(h,t)}$ と間隔個数 $N_{c(h,t)}$ から (9) 式によって求められる。また、セル $c(h,t)$ のバーストの強さを表現するバースト性 [33] を (10) 式と定義する。

$$avg(c(h,t)) = \frac{G_{c(h,t)}}{N_{c(h,t)}} \tag{9}$$

$$brt(c(h,t)) = \frac{avg(c(h,t))}{avg(c(n-1, t-1-h))} \tag{10}$$

そして、バーストを判定するパラメータ $\beta (0 < \beta < 1)$ を設定し、閾値として (11) 式が満たされる時、バーストと判断する。

$$brt(c(h,t)) \leq \beta \tag{11}$$

次に過剰なバースト検出を抑制するためにパラメータ A_{min} を設定する。リアルタイムバースト検出手法ではパケット到着間隔の変化した割合で検知を行うが、これではパケット数が少ないときでもパケット到着間隔が大きく変化していれば攻撃であると判定してしまう。過剰な攻撃判定を防ぐため、間隔個数が十分に多い時に判定処理を行うようにする。具体的には、バースト判定処理は $N_{c(h,t)} \geq A_{min}$ を満たした場合に行うようにする。

5.2 提案手法

本手法は、RTB 手法の監視イベントをパケット到着とすることで DDoS 攻撃検知に適用する。しかしながら、RTB 法をそのまま攻撃検知に適用するだけでは、パケット量が多い通信に対しても攻撃であると誤検知してしまう可能性が高い。そこで、提案手法では RTB 手法のセルに対して DDoS 攻撃検知に有効な特徴を持つエントロピー値とフローサイズの拡張を加え、誤検知率の軽減を考えるとともに、効率的な集約処理を利用してエントロピー値のリアルタイム性を改善する。

本手法では、攻撃検知に利用するデータとして、フロー ID によるエントロピー値 $E_{c(h,t)}$、フローサイズヒストグラムの最大値 $F_{c(h,t)}$ を新たにセルに保持するようにする。フロー ID とは、送信元/宛先 IP アドレス、送信元/宛先ポート番号、プロトコルで区別される ID で、フローサイズとは、そのウィンドウサイズ内でのそのフロー ID の出現数である。そして、マハラノビス距離を用いて、この 2 つの特徴量について学習データとの距離を計算し、異常度の評価を行うようにする。

IP アドレスやポート番号ではなくフロー ID を利用した理由は、IP アドレスとポート番号の出現頻度をともに監視することにある。DDoS 攻撃時、DDoS 攻撃ツールを用いる場合には送信元 IP アドレスや送信元ポート番号がランダム化されて送られることがあり、攻撃時にはこのフロー ID の出現頻度が分散することが考えられる。

また、フローサイズヒストグラムの最大値を利用した理由は、通常時に比べて新規に出現するフローが多くなる可能性が高いことから、その多くがフローサイズ 1 に集中することが予想されるためである。通常時には短期間のうちに同じフロー ID が出現しやすいためフローサイズヒストグラムの分布は分散すると考えれるが、攻撃時には送信元 IP アドレスや送信元ポート番号が変化することによってフローサイズ 1 に集中すると考えられる。攻撃時にはさらにパケット量も通常時に比べて大きくなると考えられるため、最大値を抽出することによって、集中の激しさを定量化できる。

本手法ではこれら 2 つの DDoS 攻撃に特有な特徴量を用いて攻撃検知を行う。$E_{c(0,t)}$ については (12) 式にレベル 0 セルの生成式、(13) に、レベル 1 以降のセルの集約式を示す。

$$E_{c(0,t)} = -\sum_{i=1}^{n} p_i \log p_i \tag{12}$$

$$E_{c(h,t)} = E_{c(0,t)} + E_{c(h-1,t-1)} \tag{13}$$

なお、p_i はホストごとの出現確率である。この式の意味するところとしては、頂点セルデータでは、保持する区間 $L \times W_{min}$ の間のエントロピー値の合計値が保存されることになる。これによって、ウィンドウサイズを小さくしながら、ノイズの影響を軽減することが可能となる。

$F_{c(h,t)}$ について、(14) 式にレベル 0 セルの生成式、(15) 式にレベル 1 以降のセルの集約式を示す。

$$F_{c(0,t)} = FSD(\boldsymbol{x}) \tag{14}$$

$$F_{c(h,t)} = \max\{F_{c(0,t)}, F_{c(h-1,t-1)}\} \tag{15}$$

なお、$FSD(\boldsymbol{x})$ はヒストグラムデータ $\boldsymbol{x}$ の最大値を返す関数である。この式から、頂点セルデータでは、保持する区間のうち最大のフローサイズを保持することになる。

5.2.1 攻撃検知方法

提案手法では攻撃開始判定状態、攻撃継続判定状態の 2 つの検知状態を持つ。まず攻撃開始判定では、前述のバースト検知手法を利用し、(10) 式を満たす場合は攻撃開始の兆候があるとして攻撃継続判定状態に遷移する。

攻撃継続判定では頂点セルの 2 次元のベクトルデータ $E_{c(L-1,t)}$、$F_{c(L-1,t)}$ について、マハラノビス距離 M を求める。マハラノビス距離とは、(16) 式で与えられる、各データの分散が考慮された距離である。

$$M = (\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \tag{16}$$

なお、ここで Σ^{-1} は分散共分散行列の逆行列で、精度行列とも呼ばれる。

提案手法では、マハラノビス距離 M が閾値 d 以上であれば攻撃トラフィックが到達していると判定し、逆に閾値以下であれば攻撃が終了したと判定して再度攻撃開始検知処理に移行する。この時、攻撃継続判定により攻撃が継続していると判断された期間に観測された送信元 IP アドレスについて、被疑クライアントとしてマークする。ここで、攻撃が継続しているかどうかをリアルタイムに判定する理由は、正規のユーザに影響を与えないよう実施中の規制を速やかに解除するため [34] である。

攻撃の継続判定には間隔個数も利用する。(11) 式を満たした時の $brt(c(h,t))$ の逆数を α とした時に、(17) 式を満たした回数 N_{end} が $\frac{L}{2}$ 以上となった際にも、攻撃が終了したと判定するようにする。

$$brt(c(h,t)) \geq \alpha \tag{17}$$

5.2.2 学習方法

攻撃検知にマハラノビス距離を利用するため、異常度が小さいと見込まれる学習データから平均・分散のパラメータを算出する必要がある。また、ネットワークトラフィックの状態によって通常時のエントロピー値も変わるので、追従性のためにも直近のデータを用いて学習を行う必要がある。本手法ではバースト通信を極端な変化の現れる通信トラフィックであるとみなし、バーストとして判定されない期間のデータを利用してマハラノビス距離で用いる平均と分散の値を計算して（図 9)、極端なバースト通信による影響をできるだけ排除するようにする。学習用データが $C \times N_{c(h,t)}$ だけ蓄積された後に、平均および分散を計算する。ここで C は学習の頻度およびデータ数を調整するパラメータである。

一方、バースト通信が終了した直後のセルデータにはバースト通信の特徴が頂点セルデータに残存してしまう。このデータを学習データとして用いないようにするため、バーストが発生した際のバースト終了時刻を T_{end} としたときに、$T_{end}+L\times W_{min}$ 以降に生成されたセルから再度学習用データとして保持するようにする (図 10)。

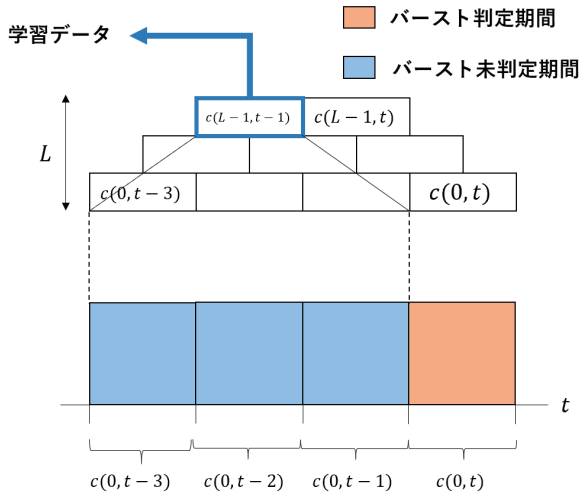


図 9. 学習用トラフィックの図

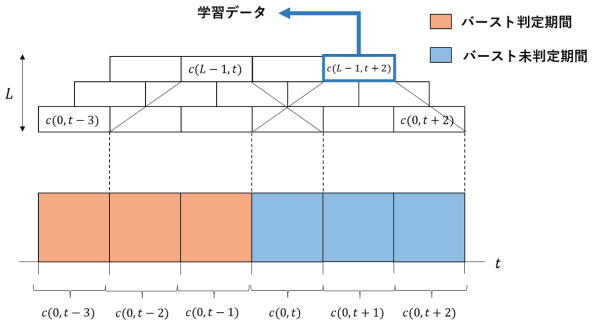


図 10. 学習データとして利用しない頂点セルの例

表 1. 実験環境

OS	Ubuntu 16.04.1
メモリ	16GB
クロック速度	3.6GHz
開発言語	C++

6. 評価実験

本章では、提案手法の性能評価として、提案手法を性能評価用のデータセットに適用し、検知精度と処理性能を調査する。実験環境は表 1 に示す通りである。

本章では利用するデータセットを述べたのち、検知精度と処理性能の調査方法について詳しく説明する。

6.1 利用したデータセット

6.1.1 DARPA2000[35]

DARPA2000 は MIT Lincoln Laboratory が人為的に作成した DDoS 攻撃のデータセットである [35]。このデータセットは 2000 年のものであるため古いものとなっているが、DDoS 攻撃検知の研究では評価に広く利用されており、既存手法との

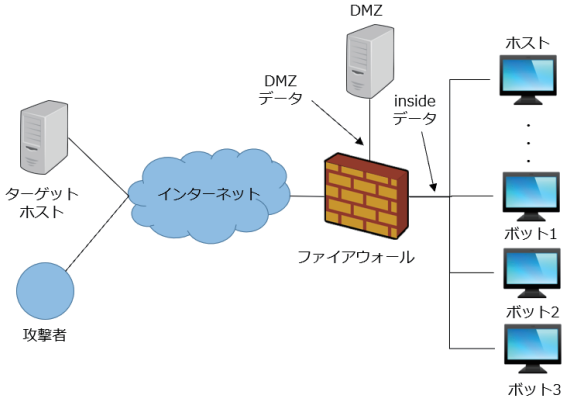


図 11. DARPA2000 のネットワーク図

比較を行えるという点で有用である。DARPA2000 は組織に侵入しホストをボット化した後に外部組織に DDoS 攻撃を仕掛けるまでのキャプチャデータが保存されており、DDoS 攻撃の他にも、IP スキャン、侵入などの攻撃パケットが観測されている。具体的に、DARPA2000 では以下の 5 段階のシナリオを想定している。

Phase1 リモートホストからターゲット組織に対して IP スキャンを行い、稼働しているホストを調査

Phase2 Phase1 で調査したアクティブなホストに対し sadmind の有無を確認

Phase3 sadmind の脆弱性を利用し、システムに侵入

Phase4 3 台のホストに DoS 攻撃ツールの mstream をインストールしボット化

Phase5 ボットに外部組織に対する DDoS 攻撃の開始を指示

DARPA2000 ではターゲット組織のファイアウォールの内部で観測された inside データと外部で観測された DMZ データが提供されている (図 11)。本研究では総パケット量の多い inside データ（総パケット数 649,787）を利用し、Phase5 の DDoS 攻撃部分のみを攻撃とみなして実験を行う。

6.1.2 CICIDS2017[36)37)]

CICIDS2017 は CIC(Canadian Institute for Cybersecurity) による IDS の性能評価のためのデータセットである [36)37)]。取得期間は 2017 年 7 月 3 日月曜日午前 9 時 から 7 月 7 日の午後 5 時までのデータとなっており、曜日ごとに分割されて提供されている。月曜日には攻撃が含まれないが、火曜日から金曜日までのデータには人為的に作成された攻撃トラフィックが記録されている。本研究では DDoS 攻撃の含まれる金曜日のインバウンドのパケットデータを利用して攻撃検知性能を調査する。DDoS 攻撃では外部の 3 台の Windows8.1 マシンから DDoS 攻撃のツール Low Orbit Ion Canon により攻撃が行われている（図 12）。特にこのデータセットは DDoS 攻撃よりも規模の大きい通常通信が観測されており、主に False Positive を調査するために利用する。

本データセットの特徴的な点として、攻撃ホストがすべて 172.16.0.1 に NAT 変換されていることが挙げられる。そのため、本手法のようにヘッダデータのみを対象に攻撃を解析する手法では DDoS 攻撃としての特徴が出ず、DoS 攻撃の様態となっている。

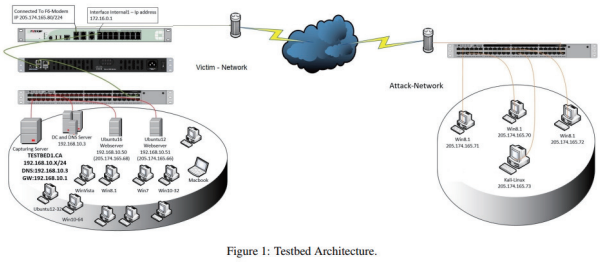


Figure 1: Testbed Architecture.

図 12. CICIDS2017 のネットワーク図 文献 [36)] より引用

6.2 検知精度

本研究の検知精度は前述の DARPA2000 と CICIDS2017 のデータセットを用いて評価する。ここでは検知精度の指標を説明した後、実験の手順を説明する。

6.2.1 指標

攻撃を正しく攻撃として検知できた数を TP (True Positive)、攻撃でないパケットを正しく攻撃でないと判定できた数を TN (True Negative) 検知されたパケットのうち、実際に攻撃でなかった数を FP (False Positive)、攻撃パケットのうち正しく検知できなかった数を FN(False Negative) としたとき、次式で示す適合率 *Precision* と再現率 *Recall*、検出精度 *Accuracy* を検知精度の指標として利用する。

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

Precision が高ければ高いほど通常パケットを攻撃だと誤判定することが少なく、*Recall* が高ければ高いほど攻撃を見逃すことが少ない、性能の高い攻撃検知手法であると言える。

6.2.2 実験手順

実験では、DARPA2000 と CICIDS2017 に対して提案手法を適用し、適合率 *Precision* と再現率 *Recall*、検出精度 *Accuracy* を算出する。パラメータ設定では W_{min} を 1 秒に固定し、レベル L を 30 から 60 まで 10 刻みで変更していった。A_{min} は通常時のデータの平均パケット量を算出した。ここで、通常データとして、DARPA2000 の場合は同じ組織で観測された DARPA1999[35)] というデータセットの、DARPA2000 と同じ曜日の攻撃の含まれないデータ、また CICIDS2017 の場合は攻撃の含まれない月曜日のデータを利用し、4.2.1 で示した方法で学習データを抽出した。この学習データは、提案手法のマハラノビス距離で用いる平均・分散値の初期値の決定にも利用した。バースト判定閾値の β は 0.01、0.05 を利用した。

マハラノビス距離の閾値 d は、次に示すホテリングの T^2 法 [38)] の 4. により χ^2 分布から決定することができる。

定理 1 (ホテリングの T^2 法) M 次元正規分布 $\mathcal{N}(\mu, \sigma^2)$ からの N 個の独立標本 $x^1, \ldots, x^N$ に基づき、標本平均を $\hat{\mu}$、標

表 2. 提案手法のパラメータ

W_{min}	L	β	C
1.0	30	0.01	1
	40	0.05	
	50		
	60		

本共分散を $\hat{\Sigma}$ と定義する。この時、$\mathcal{N}(\mu, \sigma^2)$ からの独立標本を $x^{'}$ を新たに観測した時、以下が成立する。

1. $\boldsymbol{x}^{'} - \hat{\boldsymbol{\mu}}$ は、平均 $\mathbf{0}$、共分散 $\frac{N+1}{N}\Sigma$ の M 次元正規分布に従う。
2. $\hat{\Sigma}$ は、$\boldsymbol{x}^{'} - \hat{\boldsymbol{\mu}}$ と統計的に独立である。
3. $T^2 \equiv \frac{N-M}{(N+1)M}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$ により定義される統計量 T^2 は、自由度 $(M, N-M)$ の F 分布に従う
4. $N \gg M$ の場合は、$(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$ は、近似的に自由度 M、スケール因子 1 の χ^2 分布に従う。

本研究で利用するデータの次元数は $M = 2$ であるため、χ^2 分布の自由度 2、スケール因子 1、$\alpha = 0.001$ から決まる 13.8155 とした。この α の値は文献 [30] の 0.999 という攻撃検知精度を参考にした。ホテリングの T^2 法は標本データが正規分布に従っていることを仮定しているが、データセットから抽出できる学習データに限りがあるために正規分布に従うとは限らない。そこで、ホテリングの T^2 法から決めた閾値以外にも、経験的に定めた 100 においても実験を行った。

利用したパラメータ値は表 2 に示す。

6.3 処理性能

提案手法の処理性能を調査するために、まずは、DARPA2000 に対して提案手法を適用し、ウィンドウサイズごとに行われる検知処理についての平均処理時間を算出し既存の DDoS 攻撃検知手法と比較を行う。処理性能は、ウィンドウサイズごとの処理時間をプロットし、平均と分散を算出するようにした。

次に、提案手法のエントロピー計算処理の処理効率を、既存の一般的なエントロピーベース手法と比較する。この比較実験の目的は、エントロピー計算のリアルタイム性が向上しているかどうかを確認するためである。比較の際は、ウィンドウサイズを 30 から 60 まで 5 刻みで変更していく。なお利用する情報源は提案手法で用いるフロー ID とした。

実験では、エントロピーベース手法のウィンドウサイズを W_{ent} とすると、提案手法のパラメータは $L \times W_{min} = W_{ent}$ とした時の値と比較を行う。例えば $W_{ent} = 60$ の時に、提案手法の $W_{min} = 1$ と設定した場合、L を 60 と設定した時の結果と比較する。比較対象として、提案手法のうちエントロピーの算出部分のみの結果とした。これは、提案手法ではバーストが発生した場合にはマハラノビス距離の計算というエントロピー値計算とは別の処理が発生し、純粋な比較ができないためである。

表 3. DARPA2000 における検知精度 ($d = 13.8155$)

W_{min}	β	N	*Precision*	*Recall*	*Accuracy*
1.0	0.01	30	0.978	0.944	0.991
1.0	0.01	40	0.971	0.944	0.991
1.0	0.01	50	0.959	0.944	0.989
1.0	0.01	60	0.956	0.944	0.950
1.0	0.05	30	0.919	1.0	0.990
1.0	0.05	40	0.857	1.0	0.981
1.0	0.05	50	0.920	1.0	0.990
1.0	0.05	60	0.934	1.0	0.992

7. 実験結果・考察

7.1 検知精度

7.1.1 DARPA2000 における検知精度

表 3 に検知精度の結果を示す。$\beta = 0.01$ の場合には、いずれのウィンドウサイズにおいても、*Precision* が 0.95 以上、*Recall* が 0.94 以上となっている。また、*Accuracy* については 0.990 以上となっており、同様に DARPA2000 を利用した文献 [30] の最大 1.0 には劣っているが、十分な検知精度といえる。一方で、$\beta = 0.05$ の際には $\beta = 0.01$ に比べて検知精度が減少している。*Recall* が 1.0 となっていることから、攻撃の検知はできているが、攻撃でない箇所で攻撃判定を行っていることが分かる。このことから、バーストの閾値 β に関して、組織のネットワークトラフィックに応じて調整が必要であることが分かる。

DARPA2000 においてはウィンドウサイズ $L \times W_{min}$ が小さいときに攻撃検知精度が向上している傾向がみられる。これは、DARPA2000 の DDoS 攻撃部が 5 秒程と短いことと、攻撃の規模が通常時の規模に比べて明確に差があることによって、結果的にウィンドウサイズを小さくした方がすばやく攻撃の開始を捉えることができ、攻撃の見逃しを少なくできたためであると考えられる。

また、閾値 d が 100 である時には、検知精度が向上していることが分かる。このことから、マハラノビス距離の閾値については、ホテリングの T^2 法からの閾値の妥当性が低くなっていることが考えられる。この原因については、学習に利用できるデータ数 $C \times L$ の調整係数を、データセットのデータ数の都合により $C = 1$ と設定したために、正規分布に近づくほど学習データを揃えられなかったことが考えられる。また、この場合には d を適切な値に調整する機能が必要であると言える。

7.1.2 CICIDS2017 における検知精度

表 5、6 に検知精度の結果を示す。結果としては、*Precision* が最大で 0.790 となっており、DARPA2000 での結果に比べて全体的に低くなってしまっていることが分かる。このように攻撃検知精度が低くなってしまった理由は、*Precision* と *Recall* のうちの方が小さくなっていることから、攻撃を過剰に判定してしまっていることが言える。また、DARPA2000 では攻撃観測時に攻撃トラフィックが支配的になっているが、CICIDS2017 では攻撃観測期間においても通常トラフィックが多く含まれて

表 4. DARPA2000 における検知精度 ($d = 100$)

W_{min}	β	N	*Precision*	*Recall*	*Accuracy*
1.0	0.01	30	0.978	0.944	0.991
1.0	0.01	40	0.971	0.944	0.991
1.0	0.01	50	0.959	0.944	0.989
1.0	0.01	60	0.956	0.944	0.989
1.0	0.05	30	0.961	1.0	0.995
1.0	0.05	40	0.935	1.0	0.992
1.0	0.05	50	0.959	0.944	0.989
1.0	0.05	60	0.934	1.0	0.992

表 5. CICIDS2017 における検知精度 ($d = 13.8155$)

W_{min}	β	N	*Precision*	*Recall*	*Accuracy*
1.0	0.01	30	0	0	0.953
1.0	0.01	40	0	0	0.959
1.0	0.01	50	0	0	0.950
1.0	0.01	60	0	0	0.945
1.0	0.05	30	0.438	0.999	0.609
1.0	0.05	40	0.289	0.999	0.443
1.0	0.05	50	0.294	0.999	0.455
1.0	0.05	60	0.294	0.999	0.455

いるため、DDoS 攻撃観測期間に出現したホストを攻撃だと判定する提案手法では、攻撃観測期間の抽出を正確に行えても精度が低くなってしまう。図 13、図 14 において、赤色で塗られた期間が DDoS 攻撃が観測されている期間を示しており、緑色で塗られた期間が提案手法が検知した期間を示しているが、この期間が多く重なっていることから、パラメータを適切に調整すれば DDoS 攻撃の期間自体は抽出できていることが分かる (図はパラメータ値 $W_{min} = 1.0, \beta = 0.05, L = 60, d = 100$ のとき。なお、DDoS 攻撃期間は $t = 22,972$ から $24,149$ まで、攻撃検知期間は $t = 22,994$ から $24,207$ となっている)。さらに、このとき、前半部に観測される DDoS 攻撃よりもトラフィック量の多いファイルのアップロード/ダウンロード通信を攻撃として判定していないことから、DARPA2000 のように DDoS 攻撃部のみに大量トラフィックが観測されるようなデータセットでない場合でも通常通信と攻撃通信を区別できる可能性がある。今後の課題として、ホストごとの異常度を算出できるようになれば、攻撃検知精度はより高まると考えられる。

また、$\beta = 0.01$ の時には攻撃を検知できていないことが分かる。なお、検知できていないにもかかわらず *Accuracy* が高いのは通常トラフィック量に対して攻撃トラフィック量が少ないためである。検知できなかった理由として、DARPA2000 では攻撃トラフィックの規模が平均 12,000pps となっているが、CICIDS2017 で観測された DDoS 攻撃の規模が平均 1,000pps を下回っており、バースト判定処理が行われずに攻撃判定処理が発生しなかったことが原因であると考えられる。この結果から、検知を行いたい攻撃に応じて閾値 β を調整する必要があるといえる。

表 6. CICIDS2017 における検知精度 ($d = 100$)

W_{min}	β	N	*Precision*	*Recall*	*Accuracy*
1.0	0.01	30	-nan	0	0.983
1.0	0.01	40	0	0	0.981
1.0	0.01	50	-nan	0	0.983
1.0	0.01	60	-nan	0	0.983
1.0	0.05	30	0.566	0.999	0.896
1.0	0.05	40	0.497	0.999	0.863
1.0	0.05	50	0.648	0.999	0.927
1.0	0.05	60	0.790	0.999	0.964

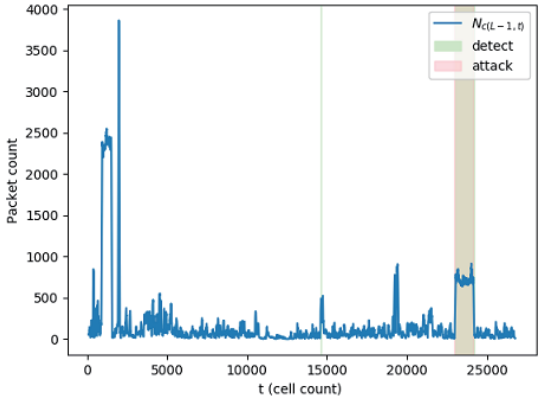


図 13. CICIDS2017 における攻撃検知期間と攻撃観測期間 (トラフィックデータ全体のグラフ。パラメータ値は $W_{min} = 1.0, \beta = 0.05, L = 60, d = 100$ で、レベル $h = L - 1$ の間隔個数 $N_(L - 1, t)$ の値を青線でプロット。

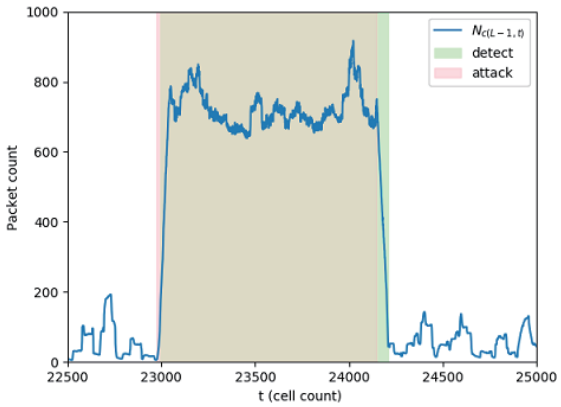


図 14. CICIDS2017 における攻撃検知期間と攻撃観測期間 (攻撃観測期間付近のグラフ。パラメータ値は $W_{min} = 1.0, \beta = 0.05, L = 60, d = 100$ で、レベル $h = L - 1$ の間隔個数 $N_(L - 1, t)$ の値を青線でプロット。

表 7. DARPA2000 におけるウィンドウサイズあたりの処理性能

ウィンドウサイズ	平均 [μsec]	分散 [μsec]
30	59.834	1668.580
40	59.5570	1631.773
50	61.380	1672.955
60	61.439	1663.542

表 8. DARPA2000 におけるエントロピー計算処理性能 (平均・分散の単位は μsec)

タイプ	ウィンドウサイズ	平均	分散
エントロピー手法	30	69.11	561.15
提案手法	30	19.01	757.729
エントロピー手法	35	72.59	604.48
提案手法	35	20.22	796.61
エントロピー手法	40	82.91	762.96
提案手法	40	19.26	750.73
エントロピー手法	45	100.53	714.30
提案手法	45	19.60	753.89
エントロピー手法	50	105.77	901.32
提案手法	50	20.29	791.60
エントロピー手法	55	114.61	824.51
提案手法	55	19.88	756.37
エントロピー手法	60	116.39	900.69
提案手法	60	19.91	756.19

7.2 処理性能

7.2.1 ウィンドウサイズごとの処理時間

表 7 に DARPA2000 におけるウィンドウサイズあたりの処理時間の結果を示す。分散値が高いのは、攻撃解析時に発生するマハラノビス距離計算が原因だと考えられる。結果から、平均としては最大でも 64.188μ 秒となっており、リアルタイム性を考慮した既存手法 [30] の 296μ 秒と比較しても処理効率が高いといえる。また、試したパラメータの範囲では、Aggregation Pyramid の階層を上げても処理効率が変わらないことが分かった。ウィンドウサイズを上げるほどノイズの影響も少なくなることから、この結果は、検知精度への影響も考えると有用であると考えられる。

7.2.2 エントロピー計算にかかる処理時間

異常検知に用いるエントロピー値の計算にかかる処理時間について、提案手法と、時間をウィンドウサイズの単位とする一般的なエントロピー手法と比較を行う。結果を表 8 に示す。結果から分散はどちらも大きいものの、平均値においてはすべてのウィンドウサイズに関して、一般的なエントロピー手法に比べて処理性能が高いことが分かる。

8. まとめ

IoT ボットネットやリフレクション攻撃の流行により、DDoS 攻撃の規模が年々増加している背景から、リアルタイムに攻撃を検知してすばやく攻撃緩和処理に移行することは重要である。

リアルタイム性を高めるアプローチとして、高速計算性を持つ手法を利用することが考えられる。エントロピー手法はパケットカウントが主な処理であるため高速計算性を持つ手法であり、DDoS 攻撃の検知精度が比較的高い特徴を持つ。一方で、エントロピー手法はノイズの影響を軽減するために広いウィンドウサイズを利用することが推奨されているが、ウィンドウサイズを大きくするとオーバヘッドが発生してしまい、処理効率が悪くなってしまう。また、検知処理の間隔が大きくなってしまうためリアルタイム性も減少してしまう。

また、エントロピー値を利用した検知では、最適な閾値の決定のために平均・分散のパラメータを学習することが重要となるが、トラフィックの状態として通常状態を定義するのは困難であることや、時間帯によってもトラフィックの状態が変わることから、異常度が少ないと見込まれる直近のデータを用いて学習する必要がある。

本手法では、効率的な集約処理を持つ、既存のデータマイニングの手法を応用して、処理間隔を小さくしながら広いウィンドウサイズで特徴量を計算するようにする。また、バーストを検出するという利点を生かし、バースト通信を除いた期間の直近のデータを逐次学習していくようにし、自動的にパラメータを決定していく。

実験結果から、検知精度は DARPA2000 の場合、*Precision* が最大 0.978、*Accuracy* が最大 0.992 となり、既存手法 [30] の *Accuracy* が最大で 1.0 という結果よりは劣るものの十分な検知精度であることが分かった。一方 CICIDSI2017 においては、検知精度として *Precision* が最大で 0.790 となっており、DARPA2000 と比較して低くなってしまっている。これはデータ数の少なさから、閾値の決定方法としてホテリングの T^2 法が適していなかった可能性があり、バースト通信期間以外のトラフィックデータを大量に収集した時に正規分布性を認められるかどうかも含めて、適切な閾値の算出方法が課題となる。

また、検知処理では既存手法 [30] の 296μ 秒よりも早くなっており、よりリアルタイム性の高い処理が可能であるといえる。また、一般的なエントロピー検知手法に比べて処理性能が高くなっており、エントロピー手法のリアルタイム性の向上が認められた。

他に挙げられる課題として、今回はデータセットのみを利用した実験しか行えていないため、実データトラフィックを用いて性能を評価する必要がある。今回利用したデータセットのパケットキャプチャファイルから十分な学習データを集めることができなかったため、ホテリングの T^2 法による自動閾値設定の有効性を確かめられるように学習データ数を増やすためにも、実データトラフィックを収集して実験を行う必要性がある。また、今回実験ではパケットキャプチャファイルをロードして攻撃検知を行っているが、現実の運用としてはオンラインキャプチャによって攻撃を検知する必要があるためオンライン実装時の性能を調査する必要がある。

参考文献

1) L. Garber: Denial-of-Service Attacks Rip the Internet, Computer, pp. 12–17, 2000.
2) P. Vixie, G. Sneeringer, and M. Schleifer: Events of 21Oct2002, http://c.root-servers.org/october21.txt (accessed 2017/02/07).
3) 中田 敦: DNS サービスの「Dyn」に大規模 DDoS 攻撃、Twitter などが影響受けダウン, https://tech.nikkeibp.co.jp/it/atcl/news/16/102203079/ (2019/02/21 閲覧).
4) Arbor 12th Annual World Infrastructure Security Report, 2017.
5) https://githubengineering.com/ddos-incident-report/ (accesed 2019/01/15).
6) Neustar: Global DDoS Attacks & Cyber Security Insights Report Persistent Threat, More Impact, 2017
7) VERSIGN: VERISIGN DISTRIBUTED DENIAL OF SERVICE TRENDS REPORT, Vol. 4, No. 1 - 1st Quarter, 2017
8) L. Munson: Greatfire.org faces daily $30,000 bill from DDoS attack, https://nakedsecurity.sophos.com/2015/03/20/greatfire-org-faces-daily-30000-bill-from-ddos-attack/ (accessed 2018/06/19).
9) G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang: An empirical evaluation of entropy-based traffic anomaly detection, presented at the Proceedings of the 8th ACM SIGCOMM conference on Internet measurement, 2008.
10) L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred: Statistical Approaches to DDoS Attack Detection and Response, Proceeding of DARPA Information Survivability Conference and Exposition, Vol.1, pp.303-314, 2003.
11) 小島 俊輔, 中嶋 卓雄, 末吉 敏則: エントロピーベースのマハラノビス距離による高速な異常検知手法, 情報処理学会論文誌 Vol.52 No.2 pp.656-668, 2011.
12) 蛯名 亮平、中村 健二、小柳 滋: リアルタイムバースト検出手法の提案, 日本データベース学会論文誌,Vol.9,No.2 pp.1-6, 2010.
13) 寺田 真敏: DoS/DDoS 攻撃とは, 情報処理学会誌, Vol. 54, No.5, pp. 428–435, 2013.
14) CDNetworks: CDNetworks, セキュリティレポート 2018 年第 1 四半期 DDoS 攻撃の動向と近穂の見通し, 2018.
15) CNET Japan, 米英蘭の法執行機関、15 の DDoS 代行サービスのドメインを差し押さえ, https://japan.cnet.com/article/35130478/ (2019/01/15 閲覧).
16) RFC 706 On the Junk Mail Problem, https://tools.ietf.org/html/rfc706 (accessed 2019/01/20)
17) 鈴木聖子:史上最大級の DDoS 攻撃に使われたマルウェア「Mirai」公開、作者が IoT を悪用,ITmedia エンタープライズ, http://www.itmedia.co.jp/enterprise/articles/1610/04/news046.html (2017/01/25 閲覧).
18) US-Cert: Alert (TA14-017A) UDP-Based Amplification Attacks https://www.us-cert.gov/ncas/alerts/TA14-017A (2014) (accessed 2019/01/07).
19) S. T. Zargar, J. Joshi, and D. Tipper: A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks, IEEE Communications Surveys & Tutorials, Vol. 15, No. 4, pp. 2046–2069, 2013.
20) 上原 孝之: 情報セキュリティスペシャリスト 2016 年版, 翔泳社, 情報処理教科書, 2016.
21) O.Osanaiye,K-KR Choo, and M. Dlodlo: Distoributed denial of Service(DDoS) resilence in cloud: Review and concenptual cloud DDoS mitigation framework, Journal of Network and Computer Applications 67, pp.147–165, 2016.
22) Snort Homepage, https://www.snort.org (accessed 2017/02/08).
23) T. Mahjabin, Y. Xiao, G. Sun, and W. Jiang: A survey of distributed denial-of-service attack, prevention, and mitigation techniques, International Journal of Distributed Sensor Networks, Vol. 13, No. 12, pp. 1–33, 2017.
24) 福田健介: インターネットバックボーントラフィックにおける異常検出, コンピュータソフトウェア, Vol. 30, No. 22, pp. 23–32, 2013.
25) C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan: A Survey of intrusion detection techniques in Cloud, Journal of Network and Computer Applications 36, pp.42–57, 2013.
26) V. Paxson and S. Floyd, Wide area traffic: the failure of Poisson modeling, IEEE/ACM Transaction. Network., Vol. 3, No. 3, pp. 226–244, 1995.
27) D. Vitali, A. Villani, A. Spognardi, R. Battistoni, and L. V. Mancini: DDoS Detection with Information Theory Metrics and Netflows - A Real Case, in Proceedings of the International Conference on Security and Cryptography, pp: 172–181, 2012.
28) M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita: An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection, Pattern Recognition Letters, Vol. 51, pp. 1–7, 2015.
29) İ. Özçelik and R. R. Brooks: Deceiving entropy based DoS detection, Computers & Security, Vol. 48, pp. 234–245, 2015.

30) N. Hoque, H. Kashyap, and D. K. Bhattacharyya: Real-time DDoS Attack Detection Using FPGA, Computer Communication., No. 11, pp. 48–58, 2017.

31) J. David and C. Thomas: DDoS Attack Detection Using Fast Entropy Approach on Flow- Based Network Traffic, Procedia Computer Science, Vol. 50, pp. 30–36, 2015.

32) X. Zhang, D. Shasha: Better Burst Detection, ICDE'06 Proceeding of the 22nd International Conference on Data Engineering, pp.146–149, 2006.

33) 蛯名 亮平、中村 健二、小柳 滋: リアルタイムバースト解析手法の提案, 情報処理学会論文誌 データベース,Vol. 5, No. 3 pp.86–96, 2012.

34) 原田 薫明, 川原 亮一, 森 達哉, 上山 憲昭, 廣川 裕, 山本 公洋: 異常トラヒック発生検出および終了判定手法, 電子情報通信関係学会, 信学技報, pp.115–120, 2006.

35) MIT:DARPA Intrusion Detection Evaluation Data Set, `https://www.ll.mit.edu/ideval/data/index.html`.

36) I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani: Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization, 4th International Conference on Information Systems Security and Privacy, 2018.

37) Intrusion Detection Evaluation Dataset (CICIDS2017), Canadian Institute for Cybersecurity, http://www.unb.ca/cic/datasets/ids-2017.html (accessed 2018/10/22)

38) 井出 剛、杉山 将: 異常検知と変化検知, 講談社, 機械学習プロフェッショナルシリーズ, 2015.