ウィンドウサイズ調節型監視による リアルタイム性DDoS攻撃検知手法の検討

溝口 李佳[1]　臼崎 翔太郎[1]　油田 健太郎[1]　山場 久昭[1]　岡崎 直宣[1]

概要：DDoS 攻撃の規模が近年増加している傾向があることから，リアルタイムに攻撃を検知することが重要である．既存の IDS は，通常ウィンドウサイズを固定し，検知を行う．しかし，ウィンドウサイズはデータの状態によって適切な値が異なると考えられており，ウィンドウサイズを固定すると，検知精度が悪化してしまったり，リアルタイムに検知できなくなったりするという問題点がある．本手法では，複数サイズのウィンドウを用意し，データの状態によって適切なウィンドウを選択し，そのウィンドウサイズで事前に学習した特徴量を用い，攻撃検知を行う．これを実現するために，複数ウィンドウ監視では Aggregation Pyramid を利用し，ウィンドウ選択では ADWIN の判定式を応用した．

キーワード：DDoS 攻撃，リアルタイム検知，異常検知

A Real-Time DDoS Attack Detection Method Using Dynamic Window Size Monitoring

Momoka Mizokuchi[1]　Shotaro Usuzaki[1]　Kentaro Aburada[1]　Hisaaki Yamaba[1]
Naonobu Okazaki[1]

Abstract: In recent years, the scale of DDoS attacks has increased. Hence, it is vital to detect attacks in real-time. Existing IDSs usually perform detection with fixed window size. However, the appropriate window size varies depending on the state of the date and fixing the window size both lowers the detection accuracy and prevents real-time detection. In our method, we prepare windows of multiple sizes and select the appropriate window depending on the data status. We then use the previously learned features in that window size for future attack detection. To achieve this, we use the Aggregation Pyramid for multiple window monitoring and the ADWIN formula for window selection.

Keywords: Distributed Denial-of-Service attack, real-time burst detection, Anomaly detection

1. はじめに

インターネットが社会インフラとなっている昨今では，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）攻撃の検知及び緩和処理が重要となっている．

DDoS 攻撃検知のための IDS（Intrusion Detection System）として，アノマリ型の NIDS（Network Intrusion Detection System）が広く利用されている．アノマリ型検知手法では，パケット到着ごとに検知処理を行うのではなく，ウィンドウサイズと呼ばれる時間単位ごとにパケットの統計情報を計算したものを利用して検知を行う．

既存の IDS はウィンドウサイズを固定することが前提となっているが，ウィンドウサイズを単一のものに固定化すると，次の問題点が考えられる．

一つ目は，データの分布が変化することを考慮しながら，適切なウィンドウサイズを設定することが難しい点である．小さいウィンドウサイズは攻撃検知速度が速く，小さい規模であったり短期間であったりする攻撃も把握するこ

[1] 宮崎大学
University of Miyazaki

とができるが，ノイズが生じることにより安定した検知が難しくなっている．一方，大きいウィンドウサイズではノイズを誤検知する可能性は減少するが，小さい規模や短期間の攻撃を検知できなかったり，攻撃検知速度が遅くなってしまったりする．このように，ウィンドウサイズの大小にはトレードオフの関係があるため，単一の値で監視する場合，分布の変化を考慮しながら適切な値に調整することが難しい．

二つ目は，攻撃者は攻撃の種類や規模を自由に組み合わせることができる点である．非常に短い期間に大規模な攻撃トラフィックを間欠的に送信し，一瞬のバースト規模を大きくしながら，平均的なトラフィック量を低減させる Pulsing Attack も観測されている [1] [2]．この攻撃では，大きなウィンドウサイズを取ると検知できなくなったり，検知が遅延してしまったりする可能性がある．

これらを踏まえると，短期間に急激に変化する異常を素早く見つけるためには，小さいウィンドウサイズでの監視が最適となる．そうでない異常を精度よく見つけるには，誤検知しないように大きなウィンドウサイズでの監視が最適となる．そのため，データの分布が変化したときに，ウィンドウサイズを適切なものに変化させる必要がある．

以前の研究では，複数のウィンドウサイズを同時監視することでこの問題に対処しようとした [3]．しかし，トラフィックデータの状態を考慮せず，単に監視するウィンドウの過半数で異常が認められた時に攻撃と判定する方式を取っていたため，最小ウィンドウと最大ウィンドウの効果が薄れることになった．

本研究では，複数のウィンドウサイズを同時監視したうえでデータの状態によって，適切なウィンドウサイズでの監視を行うようにする．バースト規模によって，攻撃は検知できるが可能な限りノイズを低減できる最低限の大きさとなるウィンドウサイズを選択する．ウィンドウサイズの同時監視は Aggregation Pyramid を利用し，適切なウィンドウサイズは ADWIN アルゴリズム [4] の判定式を応用して決定する．通常時には大きなウィンドウサイズで監視し，分布が急激に変化した場合はウィンドウサイズを小さくし，リアルタイム性を重視するようにする．

2. 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．

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 [5] 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，そのパターンと比較して合致するものを攻撃と判定する手法である．この手法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．しかし，パターンの登録数が多く，パケットが大量に到着した際には，リアルタイム性や処理時間に問題が生じる．また，未知の攻撃には対応できず，定期的に最新のパターンを反映する必要がある．

アノマリ型検知手法は，パケット単位で処理を行うのではなく，ウィンドウサイズという指定した間隔ごとに計算した特徴量を用いて異常検知を行う手法である．シグネチャ型検知手法と比較しての利点は，未知の異常を検知できる可能性がある点である．また，処理効率が高く，リアルタイム性が高い．しかし，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 [6] や，パラメータを調整しなければならない点が欠点である．さらに，一般的にウィンドウサイズを大きくするとノイズの影響は軽減できるが，検知頻度が減少するため，リアルタイム性が減少する．

アノマリ型検知手法の中で，高速計算性を持ち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．

エントロピーベース手法では，ウィンドウサイズの適切な決定の仕方が問題点となる．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 [7]，パケット数単位であれば数万程度 [8] のウィンドウサイズが推奨されている．しかし，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 [9]．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頻度を多くすることが考えられるが，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す．

このように，アノマリ型の DDoS 攻撃検知手法では，ウィンドウサイズが攻撃検知精度や攻撃検知速度などの性能に影響を与える．

3. 提案手法

3.1 アイディア

既存の多くの IDS では，ウィンドウサイズはある値に固定しておき，管理者が適宜調節するようになっている．ウィンドウサイズを固定すると，データの分布が変化した時に対応できない．

以前の研究では，L 個のウィンドウサイズでそれぞれ攻撃を判定し，攻撃と判定したウィンドウが過半数を超える場合，攻撃と判定していた [3]．本研究では，別の指標によって，一つのウィンドウサイズを L 個のウィンドウサイズの中から選択し，選ばれたウィンドウのみで攻撃判定を行う．選ばれなかった $L-1$ 個のウィンドウは攻撃判定を行わない．

これを実現させるために，提案手法は以下の二つの機能からなる．

一つ目は，複数ウィンドウサイズ監視機能である．複数

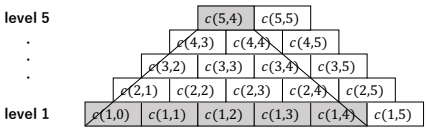

図 1: $L = 5$ の時の Aggregation Pyramid の図．$c(5,4)$ は $c(1,0)$ から $c(1,4)$ までのセルデータを集約した情報を保持している．

のウィンドウサイズのスライドウィンドウを用意し，同時に監視する．

二つ目は，ウィンドウサイズ選択機能である．攻撃検知時に，その時間のトラフィック状態によって最適なウィンドウサイズを選択し，そのウィンドウサイズで事前に学習した特徴量を用いて攻撃検知を行う．適切なウィンドウを選ぶ時には，データの分布の変化を利用し，判定する．データが変化した時に，最小サイズのウィンドウを選び，徐々に最大サイズのウィンドウに戻す．

次節から，具体的な方法を説明する．

3.2 複数ウィンドウサイズ監視機能

複数のウィンドウサイズの特徴量を計算するために，本研究では Aggregation Pyramid を利用する．

Aggregation Pyramid は複数のセルで構成されたピラミッド構造をしており（図 1），レベル 1 から L までの L 個の階層が存在する．セルは，所属する階層を l，セルの生成タイミングを t とした時，$c(l,t)$ と表現される．同じ t の値を有するセルは，同じタイミングで生成される．

最下位セル ($l = 1$) は，最小ウィンドウサイズ W_{min} ごと（ただし，直前のセル生成時刻とその次に到着したパケットの到着時刻の差が W_{min} 以上の時は該当パケットの到着時）に生成され，パケット量やエントロピー値などのトラフィックの特徴量が直接格納される．なお，t はこの時にインクリメントされる．新しくセルデータが作られると，ピラミッド構造の右側に逐次追加される．

一方，上位のレベル $l(2 \leq l \leq L)$ のセルデータは，特徴量ををを $\boldsymbol{x} = \{x_1, x_2, ..., x_n\}$，そして集約関数を $\oplus$ としたとき，式 (1) によって集約する．ここで集約関数 $\oplus$ とは，合計，最小値，最大値などがあてはまる．

$$x_{c(l,t)} = x_{c(1,t)} \oplus x_{c(l-1,t-1)} \tag{1}$$

この時, 生成に利用されるセルである $c(l-1,t-1)$ と $c(0,t)$ は，それぞれが保有するデータの期間が重ならないようになっている．例えば, 図 1 で $c(4,4)$ を生成する場合, $c(3,3)$ と $c(0,4)$ が集約に利用されるが，どちらも同じ期間を含んでいない．

ここで, 集約処理について詳細に述べる．図 2 に最小ウィンドウサイズ $W_{min} = 1.0$, $L = 3$ の Aggregation Pyramid を示す．保持する特徴量を最小ウィンドウサイズ W_{min} 以

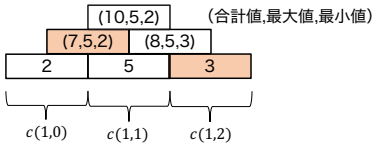

図 2: 最小ウィンドウサイズ $W_{min} = 1.0$ 秒，$L = 3$，特徴量にパケット数，集約関数に合計値，最大値，最小値を適用した例．頂点セルの $c(3,2)$ の生成に，$c(2,1) = (7,5,2)$ と，$c(1,2) = 3$ を利用している．

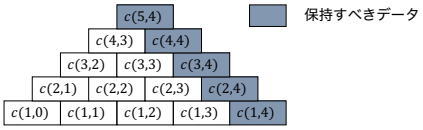

図 3: Aggregation Pyramid の形成に必要なデータ量

内に到着したパケット数とし，集約関数に合計値，最大値，最小値を適用して，それぞれのセルで保持している．今回は，図 2 の頂点セルにあたる $c(3,2)$ のデータ生成を考えることにする．$c(l-1,t-1)$ と，$c(1,t)$ にあたるセルが，オレンジ色に塗られた $c(2,1)$ と $c(1,2)$ である．$c(2,1)$ が持つデータが，合計値，最大値，最小値の順に (7,5,2), $c(1,2)$ の持つデータが 3 なので，$c(3,2)$ の保持するデータは (7+3,max{5,3},min{2,3})=(10,5,2) となる．これらの値は，$c(1,0)$ から $c(1,2)$ の間のパケット数の合計値，最大値，最小値を意味する．このようにして，レベル l セルは，l 個分の期間の特徴量を保持している．

適用する集約関数の種類を考えると, 少なくとも $l \times W_{min}$ の期間について，合計値，最大値，最小値等を効率よく監視できる．利用するデータは 2 つであるため，1 つのセルデータの更新にかかる計算量は $O(1)$ となる．この処理はピラミッドの階層数 L だけ行われるため，最終的な計算量は $O(L)$ となる．

また，最低限の時系列データを構築する上で必要なセルデータは L 個だけである．Aggregation Pyramid はピラミッド構造となっているため本来であれば $L(L+1)/2+cL$ のセルデータが必要となる．しかし，$t+1$ のセルデータを生成するのに必要なデータは，図 3 の青く塗った部分のみである．したがって，これらのセルデータさえ保持するようにすれば，新しいセルデータを生成し続けることができる．本手法ではメモリ効率を考慮してピラミッドデータをすべて保持せず，L 個分のみ保持する．

ここで，セルデータの更新処理のアルゴリズムについて Algorithm1 にまとめる．パケットの到着時刻の系列が，到着時刻を昇順にして $\boldsymbol{a} = \{a_1, a_2, ..., a_i, ..., a_n\}$ であるとき，データ構造の更新処理は Algorithm1 の手順で行われる．

Algorithm 1 Aggregation Pyramid の生成方法

```
t ← 0
while 次のパケットが存在する do
  TOP ← max(t, L − 1)
  if a_n − a_0 < W_min then
    特徴量 x_tmp の計算のためのトラフィックデータ収集
    continue
  end if
  for l = 1 to TOP do
    if l = 1 then
      a_0 から a_{n−1} までのデータを用いて x_tmp を計算
      x_{c(1,t)} ← x_tmp
    else
      x_{c(l,t)} ← x_{c(1,t)} ⊕ x_{c(l−1,t−1)}
      ウィンドウサイズ選択
    end if
    l+ = 1
  end for
  a_0 ← a_n
  t + +
end while
```

3.3 ウィンドウサイズ選択機能

複数ウィンドウ監視において，ウィンドウの数 L 個に対して，同じ数の検知器を構築している．以前の研究では各検知器でバースト判定された時に攻撃区間として検知していた．しかし，実質的に小さなウィンドウサイズと大きなウィンドウサイズの効果が無視されることになった．本研究では，ADWIN アルゴリズム [4] で利用されている判定式を応用し，L 個のウィンドウの中から現在の状態に最も適切なウィンドウサイズを決定する．

ここで，ADWIN アルゴリズムについて簡潔に述べる．このアルゴリズムは機械学習を行う上で，データの分布が途中で変化してしまうこと（Concept Drift）の対策として考案された手法である．ADWIN では新たにデータが到着すると，ウィンドウのサイズを伸ばし，ウィンドウ全体の平均と分散を更新し，Cut Detection 処理によりウィンドウ内でデータの分布が異なる区間が無いかどうか確かめる．具体的には，ウィンドウ全体を W_0 と W_1 の二つのサブウィンドウに分割して，二つのサブウィンドウ間で平均値が有意に異なっているかを確かめる．サブウィンドウの分け方は，順方向に全てのパターンで試行する．もし，平均値が有意に異なっていると判断された場合は，過去のデータと現在のデータで分布が変化したと判定し，古い方のサブウィンドウ W_0 を削除する．

ADWIN では入力データ x_t が $[0, 1]$ の間となる分布を仮定していた．しかし本研究では，それぞれの特徴量の最大値 x_{max} と最小値 x_{min} とし，$[x_{min}, x_{max}]$ の間となる分布を仮定した．ウィンドウ全体の大きさを n，W_0 のウィンドウの大きさを n_0，W_1 のウィンドウの大きさを n_1，W_0 で観測されたデータの平均値を $\hat{\mu}_{W_0}$，W_1 で観測されたデータの平均値を $\hat{\mu}_{W_1}$，信頼度を δ とすると，ヘフディングの不等式より式 (2) が成り立つ．$\hat{\mu}_{W_0}$ と $\hat{\mu}_{W_1}$ が等しいと仮定すると，ヘフディングの不等式により $\hat{\mu}_{W_0}$ と $\hat{\mu}_{W_1}$ の差がカット閾値 ϵ_{cut} を超える確率が非常に稀であるため，この閾値を超えている場合は平均値に有意な差があることが成り立つ．そのため，$\hat{\mu}_{W_0}$ と $\hat{\mu}_{W_1}$ の差が ϵ_{cut} 以上である場合に，過去と比べて現在のデータの分布が変化したと言える（式 (3)）．

$$P_r[|\hat{\mu}_{W_1} - \hat{\mu}_{W_0}| \geq \epsilon_{cut}] \leq \frac{\delta}{n} \tag{2}$$

$$|\hat{\mu}_{W_1} - \hat{\mu}_{W_0}| \geq \epsilon_{cut} \tag{3}$$

$$\epsilon_{cut} = \sqrt{\frac{(x_{max} - x_{min})^2}{2m} \ln \frac{4}{\delta'}} \tag{4}$$

$$m = \frac{1}{1/n_0 + 1/n_1} \tag{5}$$

$$\delta' = \frac{\delta}{n} \tag{6}$$

現在の 1 から L のウィンドウサイズのセルの平均値と，過去の期間の被らないウィンドウサイズ L のセルの平均値を比較し，違いがないか判定する．ウィンドウサイズ調整には，「パケットレート」「フロー ID によるエントロピー値」，「フローサイズ」を特徴量として利用する．三つの特徴量で式 (3) を同時に満たすウィンドウのうち最も大きいサイズを採用する．もしなければ，最大サイズのウィンドウを採用する．

3.4 DDoS 攻撃検知処理

ウィンドウサイズ選択機能で適切なウィンドウサイズを選択すると，攻撃検知を行う．攻撃検知では，その区間の通信が攻撃か否かの二値分類を行う．本研究では，リアルタイムにバーストを発見するシステムを想定している．攻撃検知では，「フロー ID によるエントロピー値」，「フローサイズ」を特徴量として利用する．

エントロピーベース手法とは，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，それらの増加を監視して攻撃を検知する手法である．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．

フロー ID とは，IP ヘッダの「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」の 5 つで区別される ID であり，一般に通信

を識別するために利用されている．DDoS 攻撃検知に一般的に利用される送信元 IP アドレス・宛先 IP アドレスではなく，フロー ID のエントロピー値を採用した理由は，パケットの送信元 IP アドレスあるいは宛先 IP アドレスが同じであった場合にも，ポート番号の分散具合を監視したいためである．

ここでエントロピー値の具体的な算出方法を説明する．C と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 の出現回数とすると，時間 t におけるエントロピー値算出式を以下に示す．

$$H(t) = -\sum_{i=1}^{m} p_i \log p_i = -\sum_{i=1}^{m} \frac{n_i}{C} \log \frac{n_i}{C} \qquad (7)$$

フローサイズとは，あるウィンドウ内に出現したフロー ID の異なり数である．DDoS 攻撃ツールを用いた場合は，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るため，通常時に比べて新規に出現するフロー ID が多くなると考えた．

異常検知では，「フロー ID によるエントロピー値」，「フローサイズ」の 2 次元データに関するマハラノビス距離を算出して異常度を評価する．前研究では上記の特徴量に加え，「パケットレート」を使用し，3 次元データで異常度を検知していた．本研究でパケットレートを除いた理由は，攻撃時とパケットレートの相関性が他の二つに比べ低く，通常時も攻撃と判断することがあり，検知精度を下げる原因になっていると考えたためである．

マハラノビス距離とは，式 (8) で与えられる距離である．

$$D = (\boldsymbol{x} - \boldsymbol{\mu})^T \sum^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \qquad (8)$$

ここで，$\sum^{-1}$ は分散共分散行列の逆行列で，精度行列とも呼ばれる．精度行列を掛け合わせることによって，各データの分散の違いを考慮して正規化している．これによって，各特徴量について，異常検知の寄与率が平等になるように統合している．異常検知の文脈で考えると，マハラノビス距離は正常データが多数を占めるデータで事前学習したモデルとの距離を測ることになる．この値が大きくなればなるほど，通常時に比べて異常度が大きいと言える．

特徴量が正規分布に従うならば，ホテリングの T^2 法により，マハラノビス距離の閾値 d を χ^2 分布から決定することができるが，トラフィックデータは分布を推定することが困難と知られている [6]．そのため，現在は実用上閾値は経験的に定めるようにしている．

提案手法では，マハラノビス距離 D が異常度の閾値 d 以上であれば，攻撃トラフィックが到着していると判定し，逆に閾値以下であれば攻撃でないと判定する．攻撃と判定した時に観測した送信元 IP アドレスは，被疑クライアントとしてマークする．

また，攻撃検知処理はセルの生成時に行う．ただし，パラメータとしてステップ幅 S を導入し，レベル l が 1 あるいは S の倍数のセルでのみ処理を行うとする．

4. 評価実験

本章では，提案手法の性能を評価する．提案手法，複数ウィンドウの同時監視手法，ADWIN のみの手法の三つを比較する．いずれの手法も検知器はマハラノビス距離による異常検知手法を適用した．まずは DDoS 攻撃の検知精度について評価し，攻撃をどれだけ正確に判定できるか確認する．次に DDoS 攻撃をリアルタイムに検知できるか，攻撃開始・終了の検知速度を確認する．

実験環境は表 1 に示す通りである．

表 1: 実験環境

OS	Ubuntu 20.04.3
メモリ	8GB
クロック速度	3.6GHz
開発言語	C++
ライブラリ	libpcap

4.1 利用したデータセット

本研究では，CICIDS2017 を利用し実験を行なった．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 [10] [11]．取得期間は 2017 年 7 月 3 日月曜日午前 9 時から 2017 年 7 月 7 日金曜日午後 5 時までのデータであり，曜日ごとに提供されている．月曜日には攻撃が含まれないが，月曜日以外の曜日には攻撃が含まれている．本研究では DDoS 攻撃の含まれる金曜日のデータを利用して攻撃検知性能を評価する．なお，本システムはインバウンドパケットを監視対象にすることを想定しているため，インバウンドパケットのみをデータセットから取り出し，実験を行なっている．

4.2 検知精度

本研究では，節 4.1 に記述した CICIDS2017 のデータセットを用いて評価を行う．ここでは，検知精度の指標について説明する．

4.2.1 評価指標

攻撃を正確に攻撃と判定できた数を TP（True Positive），攻撃と判定したパケットのうち攻撃でなかった数を FP（False Positive），通常パケットを正確に通常パケットと判定できた数を TN（True Negative），通常と判定したパケットのうち攻撃であった数を FN（False Negative）とする．その時，次式で示す適合率 Pr，再現率 Rc，F 値 F を検知精度の指標として利用する．

$$Pr = \frac{TP}{TP + FP} \qquad (9)$$

$$Rc = \frac{TP}{TP + FN} \tag{10}$$

$$F = \frac{2 \times Pr \times Rc}{Pr + Rc} \tag{11}$$

Pr が高いほど通常パケットを攻撃と誤判定することが少なく，Rc が高いほど攻撃を見逃すことが少ない攻撃検知手法と判断することができる．F は Pr と Rc の調和平均で算出されるもので，両者が同等程度に高いと高くなる値である．

また，それぞれの手法の ROC（Receiver Operating Characteristic）曲線と AUC（Area Under Curve）を比較する．ROC 曲線とは，閾値を変えて，検知精度を計測していった中で，横軸を false positive，縦軸を recall にした時のグラフである．AUC とは，ROC 曲線と横軸の間の領域面積のことである．一般的に AUC が大きいほど良い検知器とされる．

提案手法は 2 値分類であるため，検知精度の高さがデータセットの異常割合に依存していないことを示す．そのため，マシューズ相関係数（Matthews Correlation Coefficient）[12] を利用する．マシューズ相関係数は [-1,1] の間の値を取る．マシューズ相関係数が大きいほど，データセットの異常割合に依存していないと判断できる．

$$\mathrm{MCC} = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP) \times (TN+FN)}} \tag{12}$$

4.2.2 実験手順

実験では，CICIDS2017 に対して提案手法を適用し，適合率 Pr と再現率 Rc，F 値 F を算出する．パラメータ設定は W_{min} を 1.0 秒に固定し，レベル L を 60，120，180，240，300 と変更した．複数のウィンドウの同時監視を用いる手法も提案手法と同様に実験した．ADWIN のみを用いる手法では，W_{min} を 1.0 秒に固定し，パラメータ M を 64，128，256，512 と変更した．M は ADWIN にあるパラメータの一つで，使用されるメモリ量やカットポイントとの距離を制御する．また，δ を 0.3，パケットカウント，フロー ID によるエントロピー値，フローサイズのそれぞれの最大値 x_{max} を 900，3.3，391 とし，最小値 x_{min} を 0，0，1 とした．

なお，マハラノビス距離計算には事前学習した平均値・分散を用いる．それらは攻撃の含まれていない CICIDS2017 の月曜日のデータを学習用データセットとして利用し計算する．学習用データセットは評価用のものと同様に，インバウンドパケットのみを取り出して利用した．

4.3 検知速度

本研究では DDoS 攻撃をリアルタイムに検知できるか，攻撃開始・終了の検知速度を検知精度評価用データと同じものを用いて評価する．

4.3.1 評価指標

攻撃観測期間と攻撃検知期間のずれを調べる．攻撃検知期間のうち，実際の攻撃観測期間と最も重なる期間が多かった期間を「最大攻撃検知期間」とする．攻撃観測期間と最大攻撃検知期間の始まりの差を「検知開始の遅れ」，攻撃観測期間と最大攻撃検知期間の終わりの差を「検知終了の遅れ」とする．検知開始の遅れや検知終了の遅れの時間が短いほど，リアルタイムに攻撃を検知できていると判断する．

4.3.2 実験手順

実験手順は検知精度と同様とする．

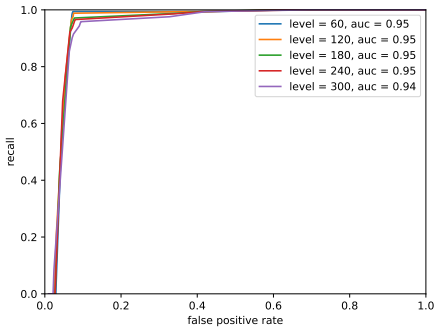

図 4: 提案手法の ROC 曲線

表 2: $W_{min} = 1.0$ での提案手法の検知精度

L	S	d	Pr	Rc	F	MCC
60	10	4	0.675	0.993	0.804	0.788
120	10	4	0.665	0.988	0.794	0.777
180	10	4	0.648	0.971	0.777	0.758
240	10	4	0.639	0.965	0.769	0.749
300	10	7	0.621	0.915	0.739	0.715

5. 実験結果

5.1 検知精度

5.1.1 ROC 曲線

まず，提案手法の ROC 曲線を図 4 に示す．

AUC は最大で 0.95 と高い結果であった．また，どの手法も AUC は 0.94 以上と高い結果であった．提案手法と複数ウィンドウの同時監視手法では，L を上げると AUC が下がったが，ADWIN のみの手法では M をあげても変化がなかった．

5.1.2 検知率

それぞれの実験結果を表 2，表 3，表 4 に示す．表には，閾値 d を 1 から 20 に 1 刻みで変更し，F 値 F が最大となったときのものを示している．

提案手法と複数ウィンドウの同時監視の手法の結果としては，どちらも $L = 60$ の時の Pr，Rc，F の全てが最大となっており，L が大きくなるにつれて検知精度が低くなっ

表 3: $W_{min} = 1.0$ での複数ウィンドウの同時監視手法の検知精度

L	S	d	Pr	Rc	F	MCC
60	10	5	0.674	0.991	0.803	0.786
120	10	5	0.671	0.985	0.798	0.781
180	10	4	0.661	0.982	0.790	0.772
240	10	6	0.656	0.969	0.783	0.764
300	10	4	0.654	0.971	0.782	0.763

表 4: $W_{min} = 1.0$ での ADWIN のみの手法の検知精度

M	d	Pr	Rc	F	MCC
64	7	0.670	0.991	0.799	0.782
128	7	0.670	0.991	0.799	0.782
256	7	0.670	0.991	0.799	0.782
512	7	0.670	0.991	0.799	0.782

ている．ADWIN のみの手法では，M を変化させても，検知精度に大きな変化はなかった．それぞれのマシューズ相関係数は 0.70 以上と高いため，データセットの異常割合に依存したものではないと考えられる．

提案手法と既存手法を比較すると，検知精度は同程度のものであった．ウィンドウサイズの選択機能を導入することによって，検知精度が大きく低下することはないと考えられる．

Pr が三つの手法で低くなっている．この結果は，DDoS 攻撃以外の通信を誤判定する割合が高かったことを表している．DDoS 攻撃だと誤検知してしまった通信は，主にポートスキャンであった．DDoS 攻撃よりもポートスキャンの通信量の方が大きかったことで，DDoS 攻撃とポートスキャンの区別をすることができなかった．DDoS 攻撃と同じ規模のポートスキャンを区別することができる検知器を導入する必要がある．

提案手法では，L が大きくなるにつれ，検知精度が低くなるという結果になった．原因としては，L が大きくなるほど，ウィンドウサイズが急激に変化しているためである．図 5 に $L = 300$ の時の提案手法における攻撃観測期間内でのウィンドウサイズの変化を示す．検知した箇所を確認すると，小さいサイズのウィンドウであれば検知できたが，大きいサイズのウィンドウを選択したために，検知が途切れてしまった．徐々に大きいウィンドウになっているが，サイズが元に戻る時間が速く，大きいウィンドウで攻撃を検知できないうちに最大のサイズに戻っている．L が大きくなるにつれ，最大サイズのウィンドウで安定して検知ができるまでに時間がかかるため，検知精度が低下していると考えられる．そのため，小さいサイズや中間サイズのウィンドウを選択した場合，一定時間ウィンドウを固定するなどして，緩やかに大きいサイズに変化させるような仕組みが必要である．

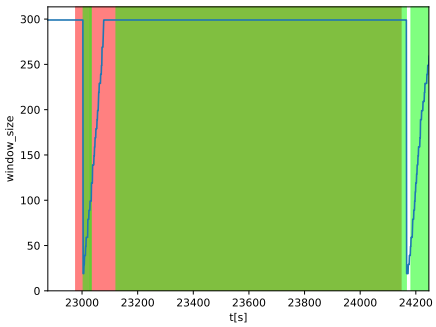

図 5: $L = 300$ の時の，提案手法のウィンドウサイズの変化．（赤が攻撃観測期間，緑が攻撃検知期間を示す．）

$L = 60, 120$ 以外は複数ウィンドウの同時監視手法の方が検知精度が高くなっている．複数ウィンドウの同時監視手法では，閾値 d を超えた数が検知処理回数の過半数である $L/2S$ 以上のときに，攻撃と判定するため，提案手法より検知精度が高いと考えられえる．

ADWIN のみの手法では，M を変化させても，変化が見られなかった．検知精度は他の二つとあまり差がなかった．

5.2 検知速度

検知速度を表 5，表 6，表 7 に示す．

提案手法では，ウィンドウサイズ選択機能によって，開始時と終了時に適切なウィンドウを選んだ結果，$L = 60$ では既存手法に比べて検知開始の遅れは短くなり，改善した．また，L を大きくしても，検知終了の遅れは 19 秒と変化しなかった．

提案手法では，$L = 60$ のときに攻撃観測期間と攻撃検知期間の差が最も短くなっている．$L = 60$ 以外では，攻撃観測期間内の検知が途切れてしまっていた．また，検知終了後に，攻撃でない期間を検知していた．これは節 5.1.2 での説明と同様に，検知開始時にウィンドウサイズを小さくし，徐々に大きいウィンドウになっているが，サイズが元に戻る時間が速く，最大サイズのウィンドウで攻撃を検知できないうちに戻っているためである．そのため，緩やかにウィンドウのサイズを戻すことによって，最大攻撃検知期間を長くすることができると考えられる．

複数ウィンドウの同時監視の手法では，攻撃の開始と終了を判定する時間が遅い．これは閾値 d を超えた数が検知処理回数の過半数以上のときに，攻撃と判定するためである．L が大きくなるにつれて，遅れる時間は大きくなると考えられる．

ADWIN のみの手法では，M を変化させても検知速度の変化は見られなかった．提案手法と比較すると，途切れなく検知がしており，安定した検知ができていると考えられる．提案手法のウィンドウの変化の仕方を改善することに

表 5: $W_{min} = 1.0$ での提案手法の検知速度

L	攻撃観測期間	最大攻撃検知期間	検知開始の遅れ [s]	検知終了の遅れ [s]
60	22977-24147	23004-24166	27	19
120		23047-24166	70	19
180		23084-24166	107	19
240		23108-24166	131	19
300		23122-24166	145	19

表 6: $W_{min} = 1.0$ での複数ウィンドウの同時監視手法の検知速度

L	攻撃観測期間	最大攻撃検知期間	検知開始の遅れ [s]	検知終了の遅れ [s]
60	22977-24147	23015-24160	38	13
120		23034-24174	57	27
180		23050-24183	73	36
240		23060-24201	83	54
300		23078-24215	101	68

表 7: $W_{min} = 1.0$ での ADWIN のみの手法の検知速度

M	攻撃観測期間	最大攻撃検知期間	検知開始の遅れ [s]	検知終了の遅れ [s]
64	22977-24147	23015-24167	38	20
128		23015-24167	38	20
256		23015-24167	38	20
512		23015-24167	38	20

よって，ADWIN のみの手法より検知速度を良くすることができると考えられる．

6. まとめ

DDoS 攻撃は，2012 年を境に年々増加しており，高精度でリアルタイム性の高い検知手法が望まれる．DDoS 攻撃検知には，アノマリ型の IDS が広く利用されている．既存の IDS は，通常ウィンドウサイズを固定し，検知を行う．しかし，ウィンドウサイズはデータの状態によって適切な値が異なると考えられており，ウィンドウサイズを固定してしまうと，検知精度が悪化してしまったり，リアルタイムに検知できなくなったりするという問題がある．

今回の手法では，複数サイズのウィンドウを用意し，データ状態に合わせて適切なサイズのウィンドウを選択し，そのウィンドウサイズで事前に学習した特徴量を用いて，攻撃検知を行う手法を検討した．これを実現するために，複数ウィンドウ監視では Aggregation Pyramid を利用し，ウィンドウ選択では ADWIN の判定式を応用した．

実験結果から，提案手法は既存手法と同程度の検知精度であり，検知遅延が少ないとわかる．そのため，提案手法では，検知精度を維持しながら，検知遅延と処理性能を改善することができたと考えられる．

今後の課題として，ウィンドウが小さくなった時にサイズを緩やかに最大に戻し，検知が途切れないようにする必要がある．そのようにすることによって，ADWIN のみの手法より検知速度を向上させることができると考えられる．

謝辞 本研究は JSPS 科研費 JP18K11268，JP21K11849 の助成を受けたものです．

参考文献

[1] A. Kuzmanovic and E. W. Knightly, “Low-rate TCP-targeted denial of service attacks: the shrew vs. the mice and elephants”, In Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM ’03, pages 75–86(2003).

[2] Igal Zeifman, “Attackers Use DDoS Pulses to Pin Down Multiple Targets”, Imperva, available at https://www.imperva.com/blog/pulse-waveddos-pins-down-multiple-targets/, 2017, (accessed 2022/02/03).

[3] 臼崎 翔太郎，油田 健太郎，山場 久昭，朴 美娘，岡崎 直宣，『複数ウィンドウサイズの効率的監視によるバースト形態に依らない DDoS 攻撃検出手法の検討』，マルチメディア，分散協調とモバイルシンポジウム 2019 論文集，pp. 495–504(2019).

[4] A. Bifet and R. Gavaldà. “Learning from Time-Changing Data with Adaptive Windowing.” In Proceedings of the 2007 SIAM International Conference on Data Mining, pp. 443–48 (2007).

[5] O.Osanaiye,K-KR Choo, 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).

[6] V. Paxson and S. Floyd, ”Wide area traffic: the failure of Poisson modeling,” IEEE/ACM Trans. Netw., vol. 3, no. 3, pp. 226-244, 1995.

[7] Vitali, Domenico, Antonio Villani, A. Spognardi, R. Trasarti Battistoni, and L. Mancini. 2012. “DDoS Detection with Information Theory Metrics and Netflows - A Real Case.” In Proceedings of the International Conference on Security and Cryptography. SciTePress - Science and and Technology Publications. https://doi.org/10.5220/0004064501720181.

[8] L. Feinstein, D. Schnackenberg, R. Balupari, D. Kindred: “Statistical Approaches to DDoS Attack Detection and Response,” Proceeding of DARPA Information Survivability Conference and Exposition, Vol.1, pp.303-314(2003).

[9] 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).

[10] 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 (ICISSP), Purtogal, January 2018

[11] Intrusion Detection Evaluation Dataset (CICIDS2017), Canadian Institute for Cybersecurity, http://www.unb.ca/cic/datasets/ids-2017.html (accessed 2022/01/21)

[12] B.W. Matthews,”Comparison of the predicted and observed secondary structure of T4 phage lysozyme”,Biochimica et Biophysica Acta (BBA) - Protein Structure,Vol. 405, No. 2, pp. 442–451, 1975.