SDN における DDoS 攻撃の 2 段階検出システムの実装と性能実証

MUYUAN NIU [1,a)]　YAOKAI FENG [1,b)]　KOUICHI SAKURAI [1,c)]

概要：近年，SDN（Software-Defined Networking）が急速に発展し，ネットワークは徐々に，従来のネットワークに取って代わられつつある．DDoS 攻撃は依然として最も危険な脅威の 1 つとなっている． 今までにも多くの検出方式が提案されているが，その基本的にはシミュレーション環境でのテストであり，検出プログラムの運用効率を分析・改良したことも少ない． 本論文では，CIC-IDS-2017 データセットを用いて，SVM 分類器に基づく 2 段階検出システムをテストして，検出性能を保ちながら計算量をなるべく大幅に削減する．また，結果の分析とこのテーマに関する継続的な課題についての考察も含まれている．

キーワード：攻撃検知，DDoS 攻撃，SDN，2 段階検出，SVM

Implementation and Performance Demonstration of a Two-Stage Detection System Using SVM for DDoS Attacks in SDN

MUYUAN NIU[†1,a)]　YAOKAI FENG[†1,b)]
KOUICHI SAKURAI[†1,c)]

Abstract: In recent years, with the rapid development of software-defined networking (SDN), the network is gradually replacing the traditional network; DDoS attacks are still one of the most dangerous threats. Although many detection methods have been proposed, they are basically tested in a simulation environment, and few of them have analyzed and improved the operational efficiency of detection programs. In this paper, we use the CIC-IDS-2017 dataset to test a two-stage detection system based on SVM classifier to reduce the computational complexity as significantly as possible while maintaining the detection performance. It also includes an analysis of the results and a discussion of ongoing challenges on this topic.

Keywords: Attack Detection, DDoS attack, SDN, Two-stage Detection, SVM

1. はじめに

SDN（Software Defined Networking）を用いてネットワークインフラを提供する SDN は，従来のネットワークの制約を克服する上で重要な役割を担っている．SDN で最も明白なのは，データプレーンとコントロールプレーンの分離である．制御プレーンは，トラフィックの送信先を決定するプレーンで，データプレーンは，この決定を実行し，実際にトラフィックを転送するプレーンである．SDN には多くの利点があるが，まだ厄介な問題が残っている．大きな課題として，SDN のセキュリティが挙げられる．SDN を標的としたサイバー攻撃には様々なものがある．中でも DDoS（Distributed Denial of Service）攻撃は非常によく知られており，SDN に最も大きな影響を与える攻撃である[1]．SDN ネットワークにおける DDoS 攻撃を検出するために，様々な再検索方法がある[2]．ネットワーク技術が進化し，ネットワークサービスへの需要が拡大する中，DDoS 攻撃の出現により，関連するネットワークサービスに異常が発生し，莫大な経済的損失やその他の破滅的な結果を誘発する可能性さえあるのである．

本論文では，DDoS 攻撃を検知するために，既存のサポートベクターマシン（SVM）アルゴリズムの拡張として，2 段階の SVM ベースの検出システムを提案する．私たちは，提案するシステムによって，3 つのリサーチクエスチョンを探求する．まず，サポートベクターマシン（SVM）が SDN の特性を活かして，どのように判定精度を向上させるかという問題である．2 つ目の問題は，SVM アルゴリズムが多くのメモリ資源を消費することである．SVM 分類器は，誤検出率が低く，高い分類精度を持つ．しかし，SVM アルゴリズムは，より多くの CPU 使用量と計算時間を必要とする．3 つ目の問題は，既存の論文の大半が実際のネットワーク環境ではなく，シミュレーションされたトラフィックのみを使用しており，オーバーフィッティングや検証不足に悩まされていることである．私たちの貢献は以下の通りである．

提案モジュールのテストケースを Mininet と Ryu のコントローラを用いて作成する．トラフィック CICD-dataset は Tcpreplay で再生される．これにより，エンドポイントは実際のネットワークデータを受信し，リアルタイムで監視されることになる．トラフィック収集時には，各スイッチ

1. Kyushu University, Japan, 744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan
a).nmy116293202@gmail.com
b).fengyk@ait,kyushu-uac.jp
c).sakurai@inf.kyushu-u.ac.jp

からトラフィックを収集し，フィーチャー生成時には，ボリュームフィーチャー，IP アドレスマッチアップ，ストリーム中の平均パケット数，ストリーム中の平均バイト数などのデータを抽出する．検出過程では，SVM の精度を維持，あるいは向上させながら，SVM アルゴリズムの呼び出し回数を大幅に削減する閾値システムを提案する．

2. 関連研究

Guo, Peng たち[3]は，データフローのエントロピーを計算し，閾値をチェックして，それを超えた場合に，そのユーザーが正当であると判断する．M.Thottan たちの論文[4]では，信号異常の良好な検出とローカライズ能力のために，ウェーブレット解析技術を適用している．まず，ネットワークの特徴量を選択し，これらの特徴量を収集して時系列を形成し，次に時系列をウェーブレット分解する．異常が発生したときのネットワークデータの変異は，ウェーブレット分解後により明らかになるため，ウェーブレット解析に基づく異常検出では，弱い異常情報を検出することができる．Navaz, A. S. たち[5]は，通常のトラフィックの基準としての歴史的なトラフィックを使用して，通常のトラフィックとして企業の特定の時間ウィンドウ内の歴史的なトラフィックを収集することにより，データの現在のウィンドウと前のウィンドウ内に収集されたデータを介してから変更，それは複雑で可変ネットワークの問題を回避する異常なトラフィックを確立することは困難であり，適応的な長さのウィンドウのアルゴリズムを提案している．

上記の研究は DDoS 検出には成功しているが，このアルゴリズムは大規模なトラフィック監視には厳しい時間と空間の複雑さが要求されるため，より多くのアプリケーションのためにさらなる改善が必要である．SDN インフラが DDoS の脅威にさらされると，コントローラ層とフォワーディング層の両方がリソースの枯渇に悩まされます．これまでの研究努力により，制御層の異常検知に多大な改善が見られるものの，詳細な分析に欠けています[6-9]．定量的な分析や実データによる裏付けが不足している。

この過剰なエネルギー消費の問題に対して，Wang, Tao たちの論文[10]では 2 段階の検出システムが設計された．トリガーモジュールを設計することで、全体の消費量を削減することができるが、検出モジュールの検討や考察は不足して、実際なネットトラフィックで実験して定量的な結果を出されていない．

3. 研究背景

3.1 SDN

SDN(Software Defined Network)，すなわち，SDN は論理的に集中したコントロール基盤と抽象化されたデータ基盤を持つ新しいネットワーク構造である．データ基盤は制御基盤から分離されており，制御基盤とデータ基盤の間には統一された開放的なインターフェースが存在する．

OpenFlow を介して，ネットワークをプログラムで直接制御することができる．

SDN ネットワークアプリケーション，ノースバンドインタフェース，SDN 制御装置，サウスバウンドインターフェース，SDNデータ基盤の 5 つの主要部分で構成されている．故に伝統的なのソフトウェアとハードウェアのネットワークと比べると以下の特徴がある．

開放的なプログラマブルなネットワーク: SDN は新しいネットワークの抽象化モジュールを確立し，ユーザーが制御装置上でネットワークの構成，制御，管理をプログラムするための共通 API の完全なセットを提供することで，ネットワークサービスのデプロイメントプロセスを加速させる．

制御基盤とデータ基盤の分離：ここでいう分離とは，制御基盤とデータ基盤の切り離しを意味する．制御基盤とデータ基盤の分離は，SDN 構造を従来のネットワーク構造と区別する重要な指標であり，ネットワークがより多くのプログラマビリティを獲得するための構造の基礎となる．制御基盤とデータ基盤の分離は，SDN 構造を従来のネットワーク構造と区別する重要な目印であり，ネットワークがより多くのプログラマビリティを獲得するための構造の基礎となる．

論理的集中管理：主に分散型ネットワークの状態を集中的に統一して管理することを指す．SDN 構造では，制御装置は，すべてのネットワークの状態情報を収集管理するという重い責任を負うことになる．論理的な集中管理は，ソフトウェアでネットワーク機能をプログラム的に定義するための構造基盤となり，ネットワークの管理を自動化する可能性を提供する．これら 3 つの特徴のうち，制御基盤とデータ基盤の分離は，論理的な集中制御の条件を作り，論理的な集中制御は，開放的なプログラマブルな制御の構造的な基盤を提供し，SDN の中核的な特徴であるネットワークの開放的なプログラマビリティを実現している．

3.2 DDoS

分散型サービス拒否攻撃（Distributed Denial of Service attack）は，多くのコンピュータを同時に攻撃し，攻撃対象を使用不能にするものである．分散型サービス拒否攻撃は，多くの大規模なウェブサイトを使用不能にすることで，ユーザーの正常な利用に影響を与えるだけでなく，莫大な経済的損失をもたらすことが何度も確認されている．

分散型サービス拒否攻撃では，攻撃元の IP アドレスを偽造することができるため，攻撃が非常に隠密に行われ，検出が困難であり，防ぐことが非常に難しい攻撃である．

Controller Plane
Flow Table

Match Rules	Counters	Actions

Ingress Port	Ether Source	Ether Dst	Ether Type	Vlan id	Vlan priority	IP src	IP dst	IP proto	IP TOS bits	TCP/UDP Src Port	TCP/UDP Dst port

表 1: フローテーブルの構造.

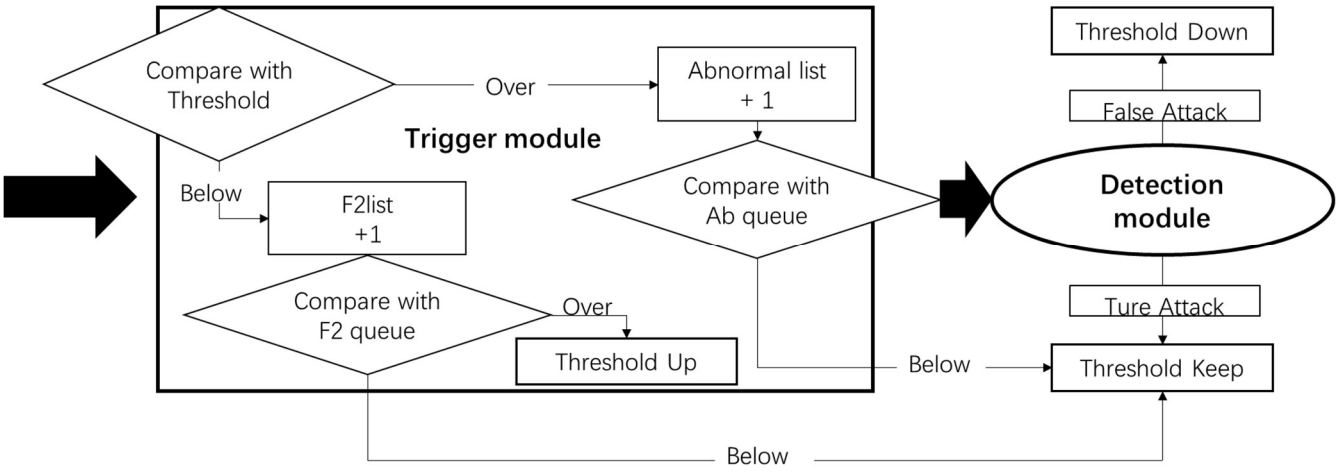

図 1 ：2 段階検出システム構造

4. 2 段階検出システム

図 2 は本提案の概念図であり，トリガーモジュールがフィールド条件を満たした場合にのみ，検出モジュールが呼び出される[10]．我々の提案は，検出モジュールがトリガーモジュールよりも多くの計算を必要とするという明白な事実に基づいている．つまり，高価な検出モジュールの起動回数を減らそうとするものである．また，第 X 章での実験結果から，本方式により，呼び出される検出モジュールの数を大幅に削減できることがわかった．さらに重要なのは，閾値が動的に調整されることである．つまり，最初に閾値を決めておけば，あとは検出・発火の状況に応じて閾値を調整すればよいのである．

4.1 トリガーモジュール

データを受信したトリガーモジュールは，既存の閾値と比較する．

1. 閾値以下の場合．この場合，検出モジュールを呼び出す必要はないが，常に閾値を下回っていると閾値が低く設定されすぎている恐れがあるため，F2list では閾値以下の連続件数が待ち行列長より多い場合に，閾値を適切に下げるように設計されている．
2. 閾値を超えた場合．DDoS 攻撃の持続時間はデータ収集間隔よりはるかに長いことが多いため，連続した閾値以上のケースの数が Abnormal キューを超えたときにのみ，呼び出しのために検出モジュールにデータが渡される．

検出モジュールのリターン結果による

1. この時点で攻撃が発生した場合，現在の閾値は検出の進行を妨げないことが証明され，変更されない．
2. 攻撃ではない場合は，現在の閾値が低すぎるため，消費電力を抑えるために閾値を上げる必要がある．

4.2 検出モジュール

4.2.1 SVM

攻撃検知は，現在の時間帯に収集したデータを分類して，現在のネットワーク状態が正常か異常かを判断する分類問題と考えることができる．分類器判定では，抽出された 6 要素の特徴量を用いて分類学習を行い，トラフィックが異常か否かを判定する．攻撃検知の基本プロセスは以下の通りである．ネットワークデータは，時間間隔に従って 6 タプルの固有値列に抽出され，サンプル列には，ネットワークの動作の 2 つの状態を表す{normal, abnormal}フラグが割り振られる．固有値サンプルのシーケンスに従って，適切な機械学習アルゴリズムを選択して検出モジュールを構築し，そのモジュールを用いて未標識の固有値サンプルを分類する[11]．この論文では，サポートベクターマシン（SVM）アルゴリズムに基づく分類学習法を選択した[12]．SVM は統計的学習理論に基づく学習法である．大量の学習データがなくても，良好な分類結果を得ることができる．SVM のカーネル関数は，高次元マッピングによる次元の

破局問題を効果的に解決し，高次元の小サンプルデータの処理能力を向上させる．SVM は，線形分離可能な最適分類超平面に由来し，その基本はの考え方は，図 3 の 2 次元のケースで説明できる．

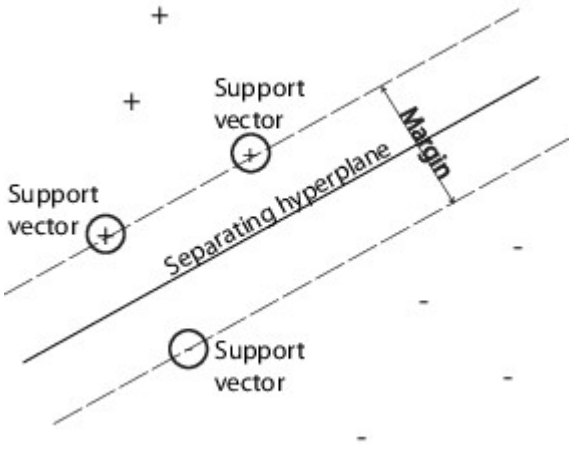

図２：分類超平面[13]

4.2.2 データ抽出

入力検出モジュールのデータは，OpenFlow switch の OpenFlow protocol から得られる．得られたデータを図 3 に示す．

```
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=0.847s, table=0, n_packets=0, n_bytes=0, idle_age=0, priority=1,tcp,in_port=1,dl_src=00:23:ae:9b:-
  8a:bf,dl_dst=00:c1:b1:14:eb:31,nw_src=192.168.10.16,nw_dst=10.0.0.2,tp_src=36958,tp_dst=80 actions=output:1
 cookie=0x0, duration=0.823s, table=0, n_packets=0, n_bytes=0, idle_age=0, priority=1,tcp,in_port=1,dl_src=00:c1:b1:14:eb:-
  31,dl_dst=00:23:ae:9b:8a:bf,nw_src=72.21.91.29,nw_dst=10.0.0.2,tp_src=80,tp_dst=36958 actions=output:1
 cookie=0x0, duration=0.796s, table=0, n_packets=1, n_bytes=66, idle_age=0, priority=1,tcp,in_port=1,dl_src=00:c1:b1:14:eb:-
  31,dl_dst=00:23:ae:9b:8a:bf,nw_src=23.54.187.27,nw_dst=10.0.0.2,tp_src=80,tp_dst=52896 actions=output:1
 cookie=0x0, duration=0.795s, table=0, n_packets=0, n_bytes=0, idle_age=0, priority=1,tcp,in_port=1,dl_src=00:23:ae:9b:-
  8a:bf,dl_dst=00:c1:b1:14:eb:31,nw_src=192.168.10.16,nw_dst=10.0.0.2,tp_src=52896,tp_dst=80 actions=output:1
 cookie=0x0, duration=1.033s, table=0, n_packets=5, n_bytes=721, idle_age=0, priority=0 actions=CONTROLLER:65535
```

図３：スイッチからのトラフィックフロー情報の一例

本論文では，SDN に関する複数の既存研究を分析・比較し，先行研究に基づくトラフィック状態情報の抽出とデータ解析・処理を行い，DDoS 攻撃の検知のために，DDoS 攻撃に関連するグラフ要素の特徴量を Source IP，Packets，Bytes の 3 観点からそれぞれ以下の 6 つ取得する．

(1) **Source IP に関して**

SDN 上で DDoS 攻撃が発生すると，攻撃者が操作したエンドポイントや，プログラムによって生成された IP スプーフィングされた IP アドレスなど，多くの異なる IP アドレスから攻撃されることが多いのである．いずれにせよ，ある時間帯にエンドポイントにアクセスするソース IP の数は，DDoS 攻撃の基準として使用することができる．

$$ASP = \frac{\sum_{i=1}^{flows} Source\_Ip}{T}$$

通常の場合，送信元ホストは宛先ホストにリクエストを送信し，応答を受信する．これは，Source_IP と Destination_IP が互いに一致するペア(Pair)が存在する，つまりインタラクティブなトラフィックを発生させる．DDoS 攻撃の場合，攻撃者からの要求に対してホストが応答することは困難であるため，Destination_IP のみがホストアドレスとなるが，ホストからの応答は存在しません．Pair の比率を計算することで，攻撃が行われているかどうかを検知する条件となる．

$$RPI = \frac{\sum_{i=1}^{n\_flows} Pair}{ASP * T}$$

(2) **Packets に関して**

AP はサンプリング間隔中の各フローのパケット数の合計である．

DDoS 攻撃の本質は，大量のパケットを送信してコントローラーを無力化することだからである．そのため，パケット数を計測することで悪意のあるトレイルを検出することができる．

$$AP = \frac{\sum_{i=1}^{n\_flows} Packets}{T}$$

SDP はサンプリング間隔におけるトラフィックパケット数の標準偏差を示す指標であり，SDN ネットワークにおける DDoS 攻撃の多くは，パケットをホストに送信して無効化することを重視する傾向があり，フルパケットは考慮せず空パケットを使用することがほとんどなので，この VPI 特性を考慮すれば，DDoS 攻撃を検出することができる．そのため，DDoS 攻撃時のパケットの標準偏差は小さくなっている．

$$SDP = \sqrt{\frac{\sum_{i=1}^{n\_flows}(Packets_i - AP)^2}{flows}}$$

(3) **Bytes に関して**

AB はサンプリング間隔中の各フローのトラフィックバイト数の合計である．AB は SDN ネットワーク上の DDoS 攻撃を検出するために使用されるが，これはほとんどの DDoS 攻撃者がパケットの送信を望んでいるためで，パケットのデータバイトは考慮されていない．したがって，トラフィックバイトの測定は，悪意のあるトラフィックを示すことができる．

$$AB = \frac{\sum_{i=1}^{nflows} Bytes}{T}$$

SDB は，サンプリング間隔内のトラフィックのバイト数の標準偏差を示す指標である．DDoS 攻撃者の多くはパケットのトラフィックバイトを考慮しないため，SDN ネットワーク上の DDoS 攻撃の検知に SDB 機能を利用することができる．そのため，トラフィックバイトの変化を測定することで，悪意のあるトラフィックを検出することができる．

$$SDP = \sqrt{\frac{\sum_{i=1}^{n\_flows}(Bytes_i - AP)^2}{flows}}$$

以上の 6 つの特徴ベクトルを用いて SVM を学習させ，取得したトラフィックをオンザフライで分類していくことになる．

5. 実験

5.1 Mininet

本研究では，VMware 上に SDN ネットワークトポロジーを作成するために，Mininet エミュレータ上で実験を行った．リュウのコントローラーを使用した．図 4 は，私たちが実装したテストベッドである．

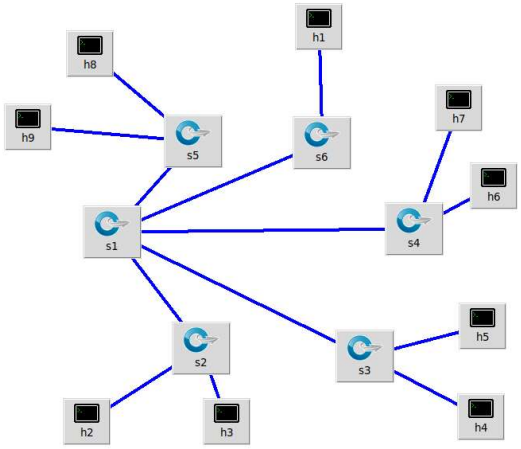

図 4 ：SDN ネットワークシミュレーション

5.2 Tcpreplay

Tcpreplay は，以前にキャプチャしたネットワークトラフィックを編集し，再生するための無料のオープンソースツールのセットです． 元々は侵入検知/防止システム向けに悪意のあるトラフィックパターンを再生するために設計されたが，ウェブサーバーへの再生機能など多くの開発が行われている[14]．この実験では，Tcpreplay に含まれる tcpwrite を使って pcap ファイルの前処理を行いリダイレクションのコマンド:

tcprewrite --infile=<filename.pcap> --outfile=<filename.pcap> --dstipmap=0.0.0.0/0:<destion IP address>

tcpreplay のコマンドを使って指定したホストでデータを再生している．その指示と結果の一部を図 6 に示す．

tcpreplay -<NIC> <host>

-x < Multi-speed playback> ./<filename>

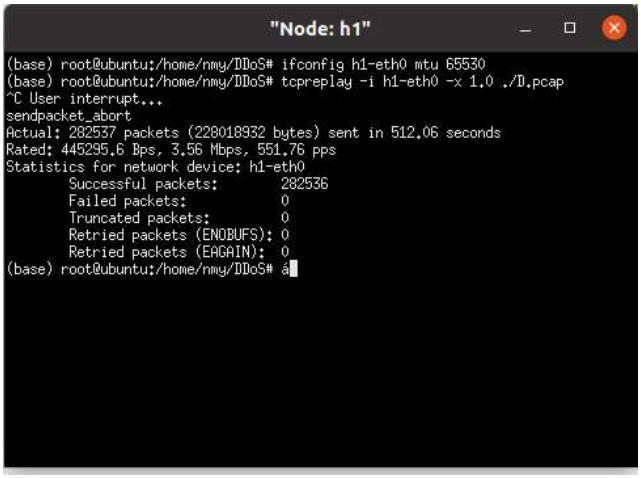

図 6 ：tcpreplay の実装例

5.3 Wireshark

Wireshark は，世界で最も多く使用されているネットワークプロトコルアナライザです．ネットワーク上で何が起こっているかをミクロのレベルで確認することができ，多くの商業・非営利企業，政府機関，教育機関においてデファクトスタンダード（しばしばデジュールスタンダード）になっている[15]．データは wireshark の editcap を使って編集し，実験中は wireshark を起動して（図 7 ）データトラフィックを監視し，実験が正しく行われたことを確認した．図 8 に示す．

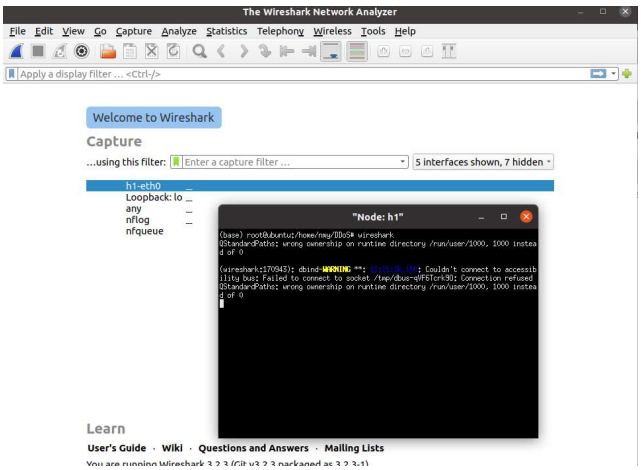

図 7 Wireshark の起動とポート選択

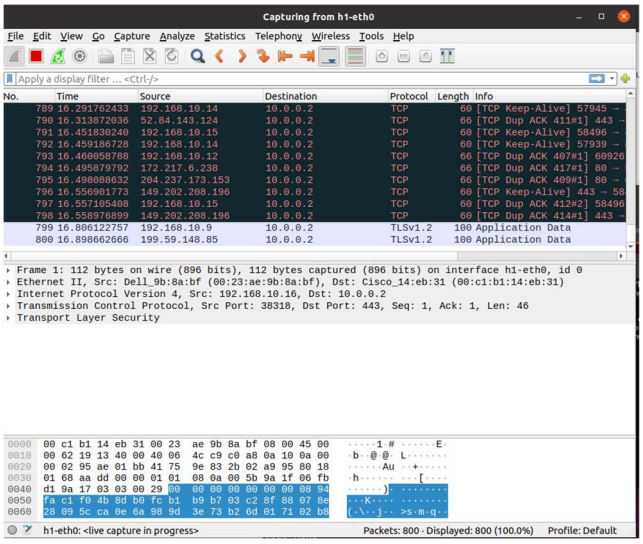

図 8 ：Wireshark リアルタイムモニタリング

5.4 目的

1. 検出モジュールの精度をテストし，攻撃を正しく区別できるかどうかを確認するとトリガーモジュールがシステム全体の検出にどのような影響を与えるかを説明する．

2. トリガーモジュールによって，IDS システム全体が全体的にリソースを消費しないようにできるかどうかをテストする．
3. 実際のデータセットを複製して，システム全体が重いネットワークトラフィックを区別できるかどうかをテストする．

5.5 データセット

CICIDS2017[16]のデータセットには，実際の実データ（PCAPs）に近い良性の攻撃と最新の一般的な攻撃が含まれている．また，CICFlowMeter によるネットワークトラフィックの解析結果や，タイムスタンプ，送信元 IP，送信先 IP，ポート，プロトコル，攻撃などを元にしたフラグ（CSV ファイル）を収録している．また，抽出された特徴量の定義も提供される．

この実験では，tcpreplay を使って CICIDS2017 のデータを再現し，2 段階の検出システムを使ってリアルタイムに検出し，得られた結果をデータのラベルと比較した．

5.6 結果

2 段階検出システムと検出モジュールは，データセット経由で 5520s のテストを行い，その結果を図 5 に示した．ここで，区別しやすいように，垂直座標 0 は通常の流れを表している．縦座標が 1 であれば，その時間に DDoS 攻撃があったことを示す．0.5 と 1.5 は，それぞれ検知モジュールが攻撃と判断したこと，2 段階検出システムが攻撃と判断したことを表している．

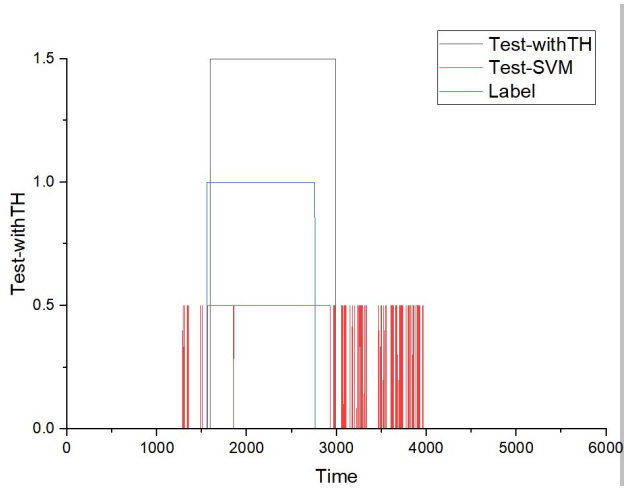

図 55520s のテスト結果

表 2，3 は，2 つのアプローチの精度，精度，感度，特異度を示したものである

表 2 RESULT(CIC-IDS2017)

	TP	FP	FN	TN
with-TH	1182	216	25	4097
SVM	1168	230	31	4091

表 3 CONFUSION MATRIX(CIC-IDS2017)

	Accuracy	Precision	Specificity	SENSITIVE
TH	0.956341	0.845494	0.949919	0.979287
SVM	0.952717	0.835479	0.946772	0.974145

表 4 は，2 段階の検出システムにおいて，検出プロセスを通じて検出モジュールが呼び出された回数と，攻撃がない場合にシステム全体と SVM 分類器の動作で消費される時間（平均値）の比較である．

表 5 時間消費量の比較

	回数	単回消費/μs
With-TH	926	813
SVM	2760	9310

5.7 考察

上述の実験結果から、次のことが分かる．

1. 上記の 6 つの特徴を用いた SVM 分類器を CICIDS2017 データセットの下でテストしたところ，精度 96%，精度 83%，感覚 95%，特異度 98%を達成することができた．また，トリガーを使って 2 段検出システムを構築しても，全体の精度は落ちず，むしろ向上した．
2. 2 段階検出方式では，SVM の呼び出し回数が大幅に減少し，トリガーモジュールの計算時間は検出モジュールの約 8%となり，CPU の計算資源消費量を削減することができた．

6. おわりに

今回は，SDN 環境における DDoS 攻撃を正確に検出しつつ，IDS システムの消費を抑えることを目的に，検出モジュールとして 6 つの特徴を持つ SVM 分類器を設計し，トリガーモジュールを追加して 2 段階の検出システムを構築している．また，実際のネットワークデータも複製してテストした．その結果，今後のトピックスを残した．トリガーモジュールにおいては，トリガーモジュールの値を設定するための具体的なスキーム．ネットワーク環境によって閾値の必要性や調整速度が異なるため，適切なパラメータを素早く選択することが効率化のための検討課題となっている．検出モジュールにおいては SVM は必ずしも 2 段検出方式に最適な検出アルゴリズムではなく，検出や学習に用いる特徴量にも改善の余地がある．

謝辞 研究に協力頂きました櫻井研究室の皆様に心から感謝致する．

参考文献

[1] Chonka, Ashley, et al. "Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks." Journal of Network and Computer Applications 34.4 (2011): 1097-1107.
[2] Jin, Xin, et al. "ZSBT: A novel algorithm for tracing DoS attackers in MANETs." EURASIP Journal on Wireless Communications and Networking 2006 (2006): 1-9.
[3] Guo, Peng, and Naixiang Li. "Self-adaptive threshold based on differential evolution for image segmentation." *2015 2nd international conference on information science and control engineering*. IEEE, 2015.
[4] M.Thottan, C.Ji.Statistical Detection of Enterprise Network Problem. Journal of Network and Systenm Managenment. Vol7, No1, 1999, 27-15
[5] Navaz, A. S., V. Sangeetha, and C. Prabhadevi. "Entropy based anomaly detection system to prevent DDoS attacks in cloud." arXiv preprint arXiv:1308.6745 (2013).
[6] Kokila, R. T., S. Thamarai Selvi, and Kannan Govindarajan. "DDoS detection and analysis in SDN-based environment using support vector machine classifier." 2014 Sixth International Conference on Advanced Computing (ICoAC). IEEE, 2014.
[7] Braga, Rodrigo, Edjard Mota, and Alexandre Passito. "Lightweight DDoS flooding attack detection using NOX/OpenFlow." IEEE Local Computer Network Conference. IEEE, 2010.
[8] Shin, Seungwon, et al. "Avant-guard: Scalable and vigilant switch flow management in software-defined networks." Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. 2013.
[9] Chen, Zhuo, et al. "XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud." 2018 IEEE international conference on big data and smart computing (bigcomp). IEEE, 2018.
[10] Wang, Tao, Yaokai Feng, and Kouichi Sakurai. "Improving the Two-stage Detection of Cyberattacks in SDN Environment Using Dynamic Thresholding." 2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM). IEEE.
[11] M. Alazab, “Profiling and classifying the behavior of malicious codes,” The Journal of Systems and Software, vol. 100, pp. 91–102, 2015.
[12] Ye, Jin, et al. "A DDoS attack detection method based on SVM in software defined network." Security and Communication Networks 2018 (2018).
[13] https://jp.mathworks.com/ Accessed on Feb. 12, 2022
[14] http://tcpreplay.appneta.com/ Accessed on Feb. 12, 2022
[15] https://www.wireshark.org/ Accessed on Feb. 12, 2022
[16] https://www.unb.ca/cic/datasets/ids-2017.html Accessed on Feb. 12, 2022