# TinyViT ベーススタッキング手法による サイバー攻撃の識別

藤本 聖己[1,a)]　青木 茂樹[1,b)]　宮本 貴朗[1]

**概要**：近年，サイバー攻撃の高度化によりネットワーク侵入検知システム（IDS）の重要性が高まっている．これまでに提案されているサイバー攻撃の識別に関する代表的な研究では，ネットワークトラフィックの TCP フローから特徴を抽出して，深層学習によりサイバー攻撃の種類を識別する手法が提案されているが，フローの途中では識別できないことが課題となっていた．そこで本研究では，単位パケット数ごとにネットワークトラフィックを画像に変換し，TinyViT ベースのスタッキング手法によりサイバー攻撃を識別する手法を提案する．具体的には，ネットワークトラフィックから IP アドレスやポート番号に基づく 3 種類の方法でパケットを抽出し，抽出したパケットを画像に変換する．次に，変換した画像を事前学習済みの TinyViT に入力し，攻撃クラスを識別するようにファインチューニングし，各画像に対して識別スコアを出力する．得られた 3 種類の識別スコアを統合したものをメタ特徴量とし，スタッキング構成のメタモデルで学習・識別を行う．実験では CICIDS2017 データセットを用いて，提案手法の有効性を確認した．

**キーワード**：サイバー攻撃の識別，IDS，TinyViT，スタッキング

# Cyberattack Classification via Stacking Ensemble of TinyViT Models

Masaki Fujimoto[1,a)]　Shigeki Aoki[1,b)]　Takao Miyamoto[1]

***Abstract:*** In recent years, the increasing sophistication of cyber attacks has heightened the importance of network intrusion detection systems (IDS). Existing studies on cyber attack identification have typically extracted features from TCP flows in network traffic and employed deep learning techniques to classify attack types. However, a significant limitation of these approaches is their inability to detect attacks during the flow. To address this issue, we propose a novel method that converts network traffic into packet-level images and applies a TinyViT-based stacking approach for cyber attack identification. Packets are extracted using three distinct methods based on IP addresses and port numbers, and the extracted packets are transformed into images. These images are then fed into a pre-trained TinyViT model, which is fine-tuned to classify attack types and outputs prediction scores for each image. The three sets of scores are integrated into a meta-feature, which is used to train a meta-model with a stacking architecture for final classification. Experimental results using the CICIDS2017 dataset demonstrate the effectiveness of the proposed method.

***Keywords:*** Cyberattack Classification, IDS, TinyViT, Stacking Ensemble

## 1. はじめに

近年，サイバー攻撃の多様化・高度化に伴ってネットワークセキュリティの確保が困難となっている．特に企業や自治体，教育機関などの情報資産が標的となる事例が増加しており，ネットワーク上の不審な振る舞いを早期に検知する仕組みである，侵入検知システム（Intrusion Detection System：IDS）の重要性が高まっている．近年特に，深層学習技術を活用した IDS の研究が活発化している．代表的な研究として，ネットワークトラフィックを TCP フロー

[1] 大阪公立大学大学院情報学研究科
Graduate School of Informatics, Osaka Metropolitan University
[a)] sd24494i@st.omu.ac.jp
[b)] aoki@omu.ac.jp

単位で処理し，深層学習によって攻撃を識別する手法 [1–4] が挙げられ，Convolutional Neural Network（CNN），Long Short Term Memory（LSTM），Transformer，AutoEncoder（AE）などのモデルにより高精度な識別を実現している．これらのフロー単位で抽出した特徴に基づく手法では，フロー継続時間やパケット間隔などの時間の特徴を活かして，高精度に攻撃を識別できる．しかし，フローの終了後にしか特徴を抽出できないため，フローの途中では識別できないという課題がある．

一方，ネットワークトラフィックを単位パケットごとに処理して，攻撃を識別する手法 [5–7] が提案されている．これらの研究では，ネットワークトラフィックデータを画像形式で表現して深層学習手法で学習し，攻撃を識別している．しかしながら，単位パケットごとの情報では通信全体の特徴を十分に捉えられないため，識別精度がフロー単位の手法よりも劣る傾向がある．

本研究では，TinyViT と Neural Network（NN）を組み合わせたスタッキング構成により，単位パケットごとに攻撃を識別する手法を提案する．まず，ネットワークトラフィックをパケットごとに，IP アドレスやポート番号に基づく 3 種類のルールで抽出し，画像に変換する．次に，変換した 3 種類の画像をベースモデルである TinyViT [8] に入力し，出力層を本研究で扱う複数の攻撃クラスの識別用に再構成してファインチューニングし，各画像に対して攻撃クラスごとの識別スコアを出力する．得られた 3 種類の識別スコアを統合したものをメタ特徴量とし，メタモデルである NN で学習・識別する．

以下，2 節で関連研究について述べ，3 節では提案手法について説明する．4 節では実験条件，実験結果，考察について述べ，5 節でまとめと今後の課題を示す．

## 2. 関連研究

本研究に関する従来研究として，TCP フロー単位の特徴と深層学習技術を用いた IDS に関する文献 [9–11] の概要を説明する．文献 [9] では，ネットワークトラフィック中の攻撃を高精度に識別するため，畳み込みニューラルネットワークと全結合型の深層ニューラルネットワークを統合した DCNN（Deep Convolutional Neural Network）モデルを提案している．この手法では，CNN で時系列特徴を抽出し，Deep Neural Network（DNN）で識別する構成となっており，両者の特性を組み合わせることで，多クラス識別および二値識別において高い性能を実現している．文献 [10] では，IoT 環境におけるネットワーク攻撃の高精度な識別を目的として，複数の深層学習モデルを統合したスタッキングアンサンブル手法 Deep Integrated Stacking for the IoT（DIS-IoT）を提案している．この手法では，Multilayer Perceptron（MLP），DNN，CNN，LSTM の 4 種類の Deep Learning モデルをベースモデルとして用い，それぞれが個別に入力データを学習した後，その出力をメタモデルである全結合層で統合して最終的な識別を行っている．DIS-IoT は，複数の公開データセットを用いた多クラス識別および二値識別の実験により，いずれのデータセットにおいても高い精度を達成している．特に，異なる識別特性を有する複数のモデルを利用することで，単一モデルよりも安定的かつ高精度に攻撃を識別できている点が特徴である．文献 [11] では，ネットワーク攻撃の識別において，データの偏りや特徴量の多さといった問題を解決するため，遺伝的アルゴリズム（GA）による特徴選択と MLP を用いたアンサンブル学習手法を提案している．まず，GA を用いて最適な特徴量と識別器の組合せを同時に探索し，選択された 4 つの識別器（ランダムフォレスト，決定木，$k$ 近傍法，サポートベクターマシン）の出力を MLP に入力して最終的な識別を行っている．実験では，不均衡なデータで高精度な識別が可能であることを確認している．

単位パケットごとに画像に変換したトラフィックデータを入力データとする深層学習技術を用いた IDS に関する文献 [6,7] の概要を説明する．文献 [6] では，VGG16 [12] と NN を組み合わせた手法を提案している．トラフィックデータを画像に変換し，VGG16 の中間層から特徴ベクトルを抽出する．そして，抽出した特徴ベクトルをニューラルネットワークで学習することで，正常な通信と異常な通信を種類ごとに識別している．この研究では，画像に現れる特徴が類似している攻撃はうまく識別できないという問題がある．文献 [7] では，サイバー攻撃の識別において，高精度かつ汎用的な識別を実現するために，ConvNeXt [13] と 1 次元畳み込みニューラルネットワーク（1D CNN）を組み合わせた手法を提案している．送信元アドレスや宛先 IP アドレス等の情報を除いたパケットのヘッダ情報を用いてトラフィックデータを画像形式に変換し，3 種類の画像化手法を用いて 32×32 画素の画像を作成する．作成した画像を用いて，ImageNet で事前学習された CNN モデルである ConvNeXt の中間層から特徴ベクトル（768 次元）を抽出して 1D CNN で学習・識別を行うことで，各トラフィックに含まれる攻撃の種類を高精度に識別している．この手法では単位パケットごとの処理を前提としているため，フロー単位の特徴を用いた手法と比較すると，特定の攻撃に対する識別精度が劣ることが課題となっていた．

## 3. 提案手法

本稿では，単位パケットごとにネットワークトラフィックを画像に変換し，TinyViT ベースのスタッキング手法によりサイバー攻撃を識別する手法を提案する．スタッキングとは，複数の機械学習モデル（ベースモデル）での識別結果を，さらに別のモデル（メタモデル）の入力として用いるアンサンブル学習手法である．提案手法の概要を**図 1** に示す．まず，ネットワークトラフィックから抽出したパ

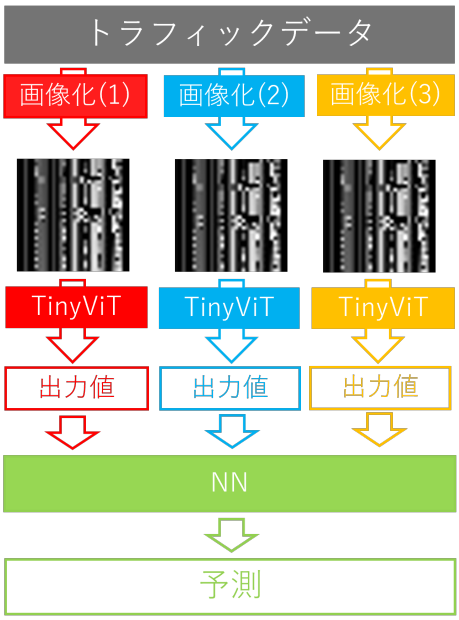


**図 1** 提案手法の概要

**Fig. 1** Overview of the proposed method.

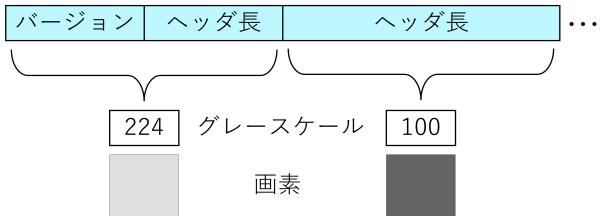


**図 2** パケットから画像への変換の概要

**Fig. 2** Overview of packet-to-image conversion.

ケットのヘッダを画像に変換する．次に，変換した画像をベースモデルに入力し，出力層を本研究で扱う複数の攻撃クラスの識別用に再構成してファインチューニングし，各画像に対して攻撃クラスごとの識別スコアを出力する．その後，得られた 3 種類の識別スコアを統合したものをメタ特徴量とし，スタッキング構成のメタモデルで学習・識別する．

### 3.1 トラフィックデータの画像変換

tcpdump 形式で取得されたトラフィックデータを，識別対象としている攻撃に適した 3 つの画像化手法により画像に変換する．ここでは，暗号化通信での攻撃の識別も想定しているため，パケットのヘッダのみを用いる．また，データ量の削減と効率化のためにパケットの内，受信パケットのみを画像変換の対象としている．まず，以下の 3 つの手法でパケットを抽出する．

(1) 宛先 IP アドレスが同一のパケットを抽出

- 特定のホストへの通信を捉え，DDoS 攻撃や Port Scan 攻撃などの識別を想定

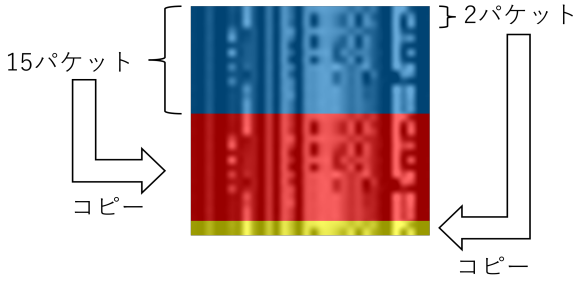


**図 3** パケット数が 32 未満の画像化の例

**Fig. 3** Example of image generation with fewer than 32 packets.

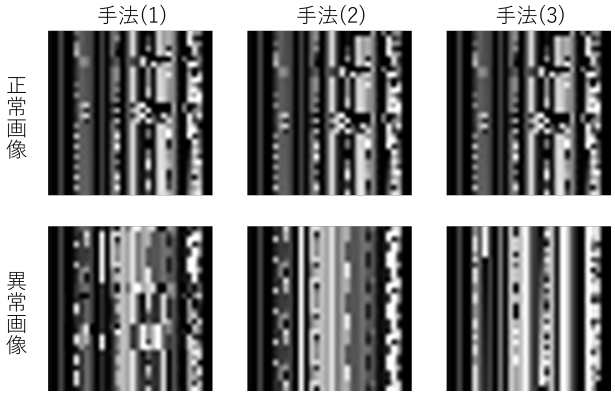


**図 4** 正常画像と異常画像の例

**Fig. 4** Examples of normal and anomalous images.

(2) 宛先 IP アドレスと宛先ポート番号の組み合わせが同一のパケットを抽出

- 特定のホストの特定のサービスへの通信を捉え，XSS 攻撃や Brute Force 攻撃などの識別を想定

(3) 宛先 IP アドレスと送信元 IP アドレスの組み合わせが同一のパケットを抽出

- 特定のホストから特定のホストへの通信を捉え，DoS 攻撃などの識別を想定

それぞれの手法で抽出したパケットを**図 2** に示す方法により画像に変換する．抽出したパケットから 1 パケット取得し，ヘッダの先頭から 8bit 単位で 0~255 の数値に変換し，変換した数値を 1 画素のグレースケールの値に割り当てる．ここで，使用しているデータに固有の情報に依存しない汎用的な IDS とするため，送信元 MAC アドレス，宛先 MAC アドレス，送信元 IP アドレス，宛先 IP アドレス，送信元ポート番号，オプションを除いている．パケットサイズが小さく，途中で割当が終了した場合は終了した箇所から 32 画素目まで白画素を割り当てる．この処理を 32 パケットに対して行い，32×32 画素の画像を 1 枚作成する．ここで，パケット数が 32 未満であった場合は，抽出したパケットを繰り返し割り当てて 32×32 画素の画像を作成する．抽出したパケットが 15 パケットであった場合の画像化の例を**図 3** に示す．まず，1 行目から 15 行目

まで抽出したパケットの情報を割り当てる．次に 16 行目から 30 行目までは，1〜15 行目と同じ情報を割り当てる．更に，31，32 行目は，1，2 行目と同じ情報を割り当てる．以上の処理で生成した正常画像と攻撃通信を含む異常画像の例を**図 4** に示す．

### 3.2 画像へのラベル付与

前節で生成した各画像にクラスを表すラベルを付与する．生成した画像に 1 パケット以上攻撃元からのパケットが含まれている場合を異常画像と定義し，攻撃の種別をラベルとして付与する．画像中に攻撃元からのパケットが存在しない場合は正常画像とし，Benign ラベルを付与する．

### 3.3 ベースモデルの学習と識別

ベースモデルには，大規模物体認識データセットで事前学習された TinyViT を用いる．TinyViT は小型かつ効率的な Vision Transformer であり，高速な蒸留フレームワークにより高い精度と転移学習能力を両立している．前節で述べた 3 種類の画像化手法で作成した画像をそれぞれ TinyViT に入力し，出力層を $C$ クラスの識別用に再構成して，前節で付与したラベルを用いてファインチューニングする．ファインチューニング後，テスト用の各画像の $C$ 次元の識別スコアを 3 種類の TinyViT で出力する．

### 3.4 メタモデルの学習と識別

前節で学習・識別した各画像に含まれるパケットは，画像化手法ごとに異なるため，メタモデルはパケット単位で学習・識別する．ベースモデルでの識別の結果，得られる $C$ 次元の識別スコアは，画像に含まれる全パケットに共通すると考える．そして，3 種類のベースモデルの $C$ 次元の識別スコアを結合して，パケットごとに $3C$ 次元のメタ特徴量を生成する．その後，注目しているパケットが攻撃元からのパケットである場合は攻撃の種類のラベルを付与し，攻撃元以外からのパケットの場合は Benign ラベルを付与する．以上の処理で生成したメタ特徴量とラベルを 2 層の全結合層で構成した NN で学習する．第 1 層では入力を 128 次元に変換し，バッチ正規化と ReLU による非線形変換を行い，ドロップアウト処理で過学習を防ぐ．第 2 層は $C$ クラスに対応する出力層である．テスト用パケットでも同様の処理を行い，出力層の最大値に対応するクラスを最終的な識別結果とする．

## 4. 実験

本手法の有効性を確認するために CICIDS2017 データセット [14] を用いて実験を行った．評価指標には F-measure を用いた．評価指標の計算式を式 (1)〜(3) に示す．ここで，TP（True Positive）は注目しているクラスを正しく注目しているクラスであると予測した数，TN（True Negative）は注目していないクラスを正しく注目していないクラスであると予測した数，FN（False Negative）は注目しているクラスを誤って注目していないクラスであると予測した数，FP（False Positive）は注目していないクラスを誤って注目しているクラスであると予測した数を表す．

**表 1** 総パケット数と各画像化手法ごとの画像数
**Table 1** Total number of packets and corresponding number of images generated by each imaging technique.

| クラス | Packets | Images | | |
|---|---|---|---|---|
| | | (1) | (2) | (3) |
| Benign | 2,182,369 | 66,919 | 122,422 | 83,973 |
| Botnet | 5,083 | 450 | 322 | 163 |
| DDoS | 744,843 | 23,276 | 23,276 | 23,276 |
| GoldenEye | 64,279 | 2,010 | 2,010 | 2,010 |
| Hulk | 1,240,352 | 38,762 | 38,762 | 38,762 |
| Slowhttptest | 32,459 | 1,015 | 1,015 | 1,015 |
| Slowloris | 37,236 | 1,165 | 1,165 | 1,165 |
| FTP-Patator | 43,323 | 1,394 | 1,355 | 1,355 |
| Heartbleed | 28,412 | 929 | 888 | 888 |
| Infiltration | 28,828 | 2,043 | 909 | 903 |
| PortScan | 162,002 | 5,063 | 6,115 | 5,063 |
| SSH-Patator | 54,678 | 1,726 | 1,709 | 1,709 |
| Brute Force | 19,643 | 621 | 615 | 615 |
| SQL Injection | 67 | 3 | 3 | 3 |
| XSS | 6,344 | 199 | 198 | 198 |

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{F-measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

### 4.1 実験条件

#### 4.1.1 実験データセット

CICIDS2017 データセットは，2017 年に Canadian Institute for Cybersecurity において収集された，現実世界のネットワークトラフィックを模倣した大規模なデータセットである．14 種類の攻撃と正常通信を含む 15 クラス構成であり，多クラス識別の検証に適している．また，パケットキャプチャデータ（tcpdump 形式），フローデータ，およびラベル付きデータが提供されており，多様な分析が可能である．今回の実験では，パケットキャプチャデータ中の正常通信と 14 種類の攻撃を含む異常通信のデータを用いて $C = 15$ とした．正常通信のデータが異常通信のデータと比べて多いため，正常通信のパケット数を異常通信の全パケット数と同等程度になるように調整している．そして，ベースモデルの学習データ，メタモデルの学習データ，テストデータを 6:3:1 に分割して実験を行った．**表 1** に実験に使用したパケット数と画像化手法ごとの画像枚数を示す．

表 2　提案手法による攻撃種別ごとの F-measure
**Table 2**　F-measure for each attack type using the proposed method.

| クラス | stacking | (1) | (2) | (3) |
|---|---|---|---|---|
| Benign | 0.9995 | 0.9955 | 0.9956 | 0.9967 |
| Botnet | 0.8779 | 0.5161 | 0.9910 | 0.7170 |
| DDoS | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| GoldenEye | 0.8230 | 0.8578 | 0.8150 | 0.8849 |
| Hulk | 0.9821 | 0.9838 | 0.9786 | 0.9847 |
| Slowhttptest | 0.7359 | 0.7191 | 0.7212 | 0.7361 |
| Slowloris | 0.8763 | 0.8517 | 0.8815 | 0.8651 |
| FTP-Patator | 1.0000 | 0.9412 | 1.0000 | 0.9927 |
| Heartbleed | 1.0000 | 0.9945 | 0.9468 | 1.0000 |
| Infiltration | 0.9492 | 0.3406 | 0.2342 | 0.1600 |
| PortScan | 0.9949 | 0.9768 | 0.9355 | 0.9789 |
| SSH-Patator | 1.0000 | 1.0000 | 1.0000 | 0.9856 |
| Brute Force | 1.0000 | 0.9344 | 0.9919 | 0.9920 |
| SQL Injection | 1.0000 | 0.0000 | 0.0000 | 0.0000 |
| XSS | 0.9766 | 0.7755 | 0.9756 | 0.7273 |

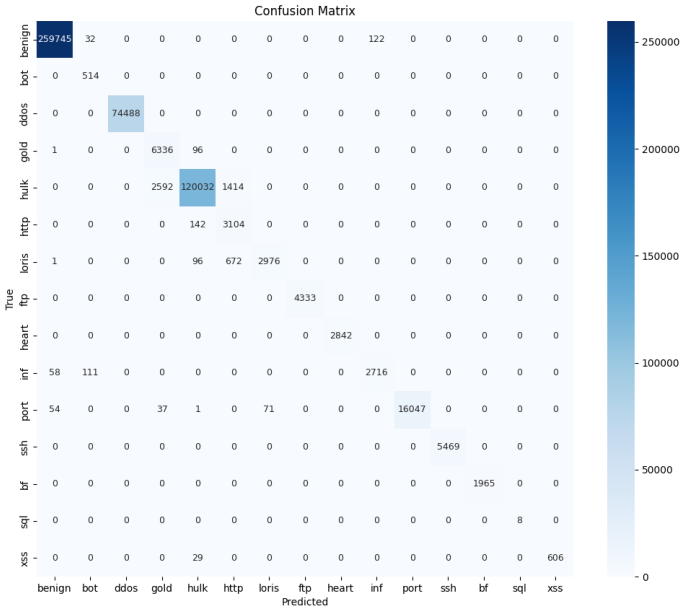


図 5　提案手法の混同行列
**Fig. 5**　Confusion matrix of the proposed method.

#### 4.1.2　ベースモデルとメタモデルの設定

ベースモデルには，TinyViT-21M/224 モデル（tiny_vit_21m_224.dist_in22k_ft_in1k）を使用した．このモデルは ImageNet-22k で事前学習され，その後 ImageNet-1k でファインチューニングされたものである．TinyViT はエポック数 10，バッチサイズ 324，学習率 0.0001，損失関数はクロスエントロピー誤差，最適化アルゴリズムは Adam を使用してファインチューニングした．メタモデルである NN はエポック数 100，バッチサイズ 4096，学習率 0.001，損失関数はクロスエントロピー誤差，最適化アルゴリズムは Adam を使用して学習した．また，過学習を防ぐために，学習が収束した時に自動的に学習を止める Early Stopping を使用した．

### 4.2　実験結果と考察

スタッキングモデルを用いる提案手法と，ベースモデルである 3 種類の画像を入力とする TinyViT のそれぞれで，攻撃ごとの識別精度を確認する実験を行った．また，提案手法の有効性を評価するために文献 [9–11] の手法と比較した．

#### 4.2.1　実験結果

実験の結果，得られた提案手法による攻撃種別ごとの F-measure を**表 2** に示す．表 2 の結果から，提案手法が多くの攻撃に対して高い識別精度を実現していることがわかる．特に F-measure が 1.0000 である 6 種類の攻撃では，完全に誤りなく識別できており，モデルがこれらの攻撃パターンの特徴を安定的に学習できていることを確認できる．一方，GoldenEye（0.8230）や Slowhttptest（0.7359），Slowloris（0.8763）の一部の DoS 系攻撃に対しては他の攻撃に比べて F-measure が低い結果となっている．

提案手法では，画像化手法 (1)〜(3) のそれぞれのベースモデルで精度がばらつくクラスにおいて，弱点を補完し合う形で高い識別精度を実現できている例が存在した．Infiltration では，ベースモデル (1)(2)(3) で低い値（0.3406，0.2342，0.1600）となっているが，スタッキング構成による統合学習を通じて 0.9391 という高い F-measure が得られていることから，3 種類のベースモデルの情報統合が有効に働いていると評価できる．また，SQL Injection はベースモデルの結果が全て 0.0000 であったにもかかわらず，スタッキングの結果が 1.0000 となっており，ベースモデルでは検知できていないが，その出力値がメタモデルの学習の有効な特徴量となったと考えられる．このような結果から，提案する TinyViT ベースのスタッキング構成は，従来の単一モデルを用いた場合と比較して，より安定かつ高精度な攻撃識別を実現していることを確認できた．

提案手法の混同行列を**図 5** に示す．図 5 に示した混同行列は，提案手法における各攻撃クラスおよび正常通信に対する識別結果を視覚的に示したものである．この図から，Benign（正常通信）を含む大多数のクラスで，対角成分が高く，非対角成分がほぼゼロであることを確認でき，提案手法が各クラスを高精度に識別できていることがわかる．表 2 に示す結果で F-measure が 0.9000 に満たなかった Botnet，GoldenEye，Slowhttptest，Slowloris の攻撃について考察する．まず，Botnet の行と列を確認すると，Recall が 1.0000 である一方，Precision が低く，多くの Infiltration が Botnet に誤識別されていることが分かる．これは，両クラスが比較的低頻度かつ断続的な通信パターンで類似した特徴を持つため，今回用いた画像化手法では明確な差異が現れにくかったためであると考えら

**表 3** 提案手法と比較手法による攻撃種別ごとの F-measure
**Table 3** Comparison of F-measure for each attack type between the proposed method and the baseline method.

| クラス | 提案手法 | 文献 [9] | 文献 [10] | 文献 [11] |
|---|---|---|---|---|
| Benign | 0.9995 | 0.9998 | 0.9921 | 1.00 |
| Botnet | 0.8779 | 0.9685 | 0.5543 | 0.86 |
| DDoS | 1.0000 | 0.9999 | 0.9964 | 1.00 |
| GoldenEye | 0.8230 | 0.9963 | 0.9828 | 1.00 |
| Hulk | 0.9821 | 0.9994 | 0.9856 | 1.00 |
| Slowhttptest | 0.7359 | 0.9968 | 0.9780 | 0.99 |
| Slowloris | 0.8763 | 0.9966 | 0.9810 | 0.99 |
| FTP-Patator | 1.0000 | 0.9993 | 0.9966 | 1.00 |
| Heartbleed | 1.0000 | 1.0000 | 0.8000 | 1.00 |
| Infiltration | 0.9492 | 1.0000 | 0.0000 | 0.88 |
| PortScan | 0.9949 | 0.9998 | 0.9244 | 1.00 |
| SSH-Patator | 1.0000 | 0.9936 | 0.9687 | 1.00 |
| Brute Force | 1.0000 | 0.9099 | 0.2209 | 0.80 |
| SQL Injection | 1.0000 | 0.5000 | 0.0000 | 0.73 |
| XSS | 0.9766 | 0.8962 | 0.0000 | 0.35 |

れる．次に GoldenEye と Slowhttptest を見ると，いずれも Recall は高いが Precision が低くなっている．Hulk をこの 2 つの攻撃として誤識別している例が多くみられた．2 つの攻撃の F-measure が低い理由は Hulk のデータ数が多く，GoldenEye，Slowhttptest のデータ数が少ないことによるデータの不均衡が原因であると考えられる．最後に Slowloris は，Recall が低く Precision が高い傾向を示し，誤識別先として Slowhttptest が多く確認された．これらはいずれも遅延型 DoS 攻撃であり，通信パターンが類似しているため識別が難しい．特に，本研究で用いた画像化手法は 32 パケット単位で処理するため，通信全体の流れを十分に反映できず，識別が難しかったと考えられる．

これらの課題に対する改善策としては，少数クラスのデータ拡張やクラス重み付けによる不均衡是正，DoS 系攻撃の特徴を強調する画像変換手法の導入などが考えられる．

#### 4.2.2 比較手法との比較

提案手法と比較手法の結果を**表 3** に示す．表 3 は，提案手法と既存の代表的な 3 つのフロー単位の IDS 手法における F-measure を各攻撃クラスごとに示している．なお，文献 [9][11] の結果は各論文に記載された数値を引用し，文献 [10] の結果は同論文に掲載された混同行列から F-measure を算出している．提案手法は多数の攻撃クラスにおいて，他の比較手法と同等以上の F-measure を達成している．F-measure が 1.0000 である 6 種類の攻撃は，他手法と比べても優位性が確認できる．特に SQL Injection においては，文献 [9] の手法では 0.5000，文献 [10] の手法では 0.0000 であるのに対し，提案手法では 1.0000 と大きく上回っている．また，Brute Force や XSS といった他手法で性能が低下しやすい攻撃に対しても，0.97 以上の F-measure を達成しており，比較手法では識別の難しい攻撃に対する優位性を確認できた．一方，GoldenEye や Slowhttptest などの DoS 攻撃に関しては，比較手法に劣る結果となっている．前節でも述べたように少数クラスのデータ拡張やクラス重み付けによる不均衡是正，DoS 系攻撃の特徴を強調する画像変換手法の導入などで改善できると考えられる．

## 5. おわりに

本研究では，ネットワークトラフィックを単位パケットごとに画像に変換し，TinyViT をファインチューニングして各画像に対する識別スコアを出力させ，これらのスコアを統合したものをメタ特徴量とし，NN を用いて学習・識別するスタッキング構成によるサイバー攻撃の識別手法を提案した．CICIDS2017 データセットを用いた評価実験の結果から，スタッキング構成による統合学習は単一の手法よりも高い識別性能を示し，従来のフロー単位の特徴を扱う手法と比較して，同等以上の識別性能を確認した．また，提案手法は単位パケットごとに識別を行うため，フローの終了まで待つ必要がないという利点がある．今後の課題としては，NN よりも学習に優れた識別器の導入，不均衡なデータの是正，および識別精度が低下した DoS 系攻撃に対する特徴強調を意識した画像化手法の検討が挙げられる．

### 参考文献


[1] Jose, J. and Jose, D. V.: Deep learning algorithms for intrusion detection systems in internet of things using CIC-IDS 2017 dataset, *Int. J. Electr. Comput. Eng.*, Vol.13, No.1, pp.1134-1141 (2023).

[2] Sayegh, H. R., Dong, W. and Al-madani, A. M.: Enhanced Intrusion Detection with LSTM-Based Model, Feature Selection, and SMOTE for Imbalanced Data, *Appl. Sci.*, Vol.14, No.2, p.479 (2024).

[3] Manocchio, L. D., Layeghy, S., Lo, W. W., Kulatilleke, G. K., Sarhan, M. and Portmann, M.: FlowTransformer: A Transformer Framework for Flow-based Network Intrusion Detection Systems, *arXiv preprint arXiv:2304.14746v1* (2023).

[4] Huang, H., Yang, J., Zeng, H., Wang, Y. and Xiao, L.: Self-Organizing Maps-Assisted Variational Autoencoder for Unsupervised Network Anomaly Detection, *Symmetry*, Vol.17, No.4, p.520 (2025).

[5] 東畑和希, 青木茂樹, 宮本貴朗：CNN で抽出した特徴量に基づく VAE によるネットワークの異常検出, 電子情報通信学会, Vol.122, No.ISEC-428, pp.269-276 (2023).

[6] 鷺坂典雅，青木茂樹，宮本貴朗：CNN で抽出したパケットの特徴に基づくネットワークの異常検出，コンピュータセキュリティシンポジウム 2021 論文集，pp.575-582 (2021).

[7] 藤本聖己，青木茂樹，宮本貴朗：ConvNeXt によるネットワークトラフィック中の異常検出，コンピュータセキュリティシンポジウム 2024 論文集，pp.1668-1675 (2024).

[8] Wu, K., Zhang, J., Peng, H., Liu, M., Xiao, B., Fu, J. and Yuan, L.: TinyViT: Fast Pretraining Distillation for Small Vision Transformers, *arXiv preprint arXiv:2207.10666v1* (2022).

[9] Shebl, A., Elsedimy, E. I., Ismail, A., Salama, A. A. and Herajy, M.: DCNN: A novel binary and multi-class net-

work intrusion detection model via deep convolutional neural network, *EURASIP J. on Inf. Secur.*, Vol.2024, No.36 (2024).
[10] Lazzarini, R., Tianfield, H. and Charissis, V.: A stacking ensemble of deep learning models for IoT intrusion detection, *Knowl.-Based Syst.*, Vol.279, p.110941 (2023).
[11] Çetin, G.: An Effective Classifier Model for Imbalanced Network Attack Data, *Comput. Mater. Continua*, Vol.73, No.3, pp.49109 (2022).
[12] Simonyan, K. and Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, *Proc. ICLR 2015* (2015).
[13] Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T. and Xie, S.: A ConvNet for the 2020s, *arXiv preprint arXiv:2201.03545* (2022).
[14] Sharafaldin, I., Habibi Lashkari, A. and Ghorbani, A. A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization, *Proc. 4th ICISSP*, pp.108-116 (2018).