Spatial Data Transformation and Vision Learning For Elevating Intrusion Detection in IoT Networks

Van-Linh Nguyen, *Member, IEEE*, Hao-Ping Tsai, Hyundong Shin, *Fellow, IEEE,* Trung Q. Duong, *Fellow, IEEE*

***Abstract*—Network intrusion detection systems (NIDS) are vital for identifying security attacks and predicting early invasion attempts, which is essential for protecting the Internet. Recently, Deep learning (DL) has made significant achievements in enhancing intrusion detection accuracy. Nevertheless, the practical implementation of high-complexity DL models is limited by the constrained computational capabilities of the Internet of Things (IoT) devices, e.g., home routers and IoT gateways. This article introduces a novel NIDS approach explicitly tailored for IoT networks, leveraging a lightweight deep learning model. During the data preprocessing phase, we use a spatially enriched data conversion technique to decrease the dimensionality of high-dimensional raw traffic variables. This helps to offset the problem of increased model complexity. Furthermore, when spatial relationships often exist in the data, we can simplify the learning architecture by utilizing state-of-the-art vision transformer techniques in the computer vision field that can substantially reduce model complexity. The experimental results indicate that the proposed method achieves outstanding accuracy up to 99.57% with high-volume traffic input. Moreover, the proposed method reaches substantial reductions in learnable parameters by 55.35% and 82.07%, along with a remarkable decrease in floating point operations (FLOPs) by 93.56% and 99.28% compared to existing studies. The outstanding achievement highlights the proposed method's ability to balance model complexity and accuracy performance, making it extremely appropriate for deployment on IoT gateways with limited resources.**

***Index Terms*—Intrusion Detection System, Vision Transformer, Internet of Things security**

I. Introduction

Distributed denial of service (DDoS) attacks have been one of the most powerful threats against online services' availability. These attacks gain worldwide attention due to their devastating impact on businesses, governments, and critical infrastructure. DDoS attacks involve flooding a target server with excessive web/service requests, sometimes peaking at staggering rates, such as the reported case of 46 million requests per second [1], rendering these services inaccessible. What exacerbates the challenge posed by these attacks is their increasing frequency and complexity. Attackers often leverage botnets of thousands of zombie IoT devices to launch the attacks, making mitigation efforts even more challenging. Intrusion detection systems (IDS) are essential methods, specifically developed to promptly detect and coordinate to filter the volume of attack traffic [2], [3].

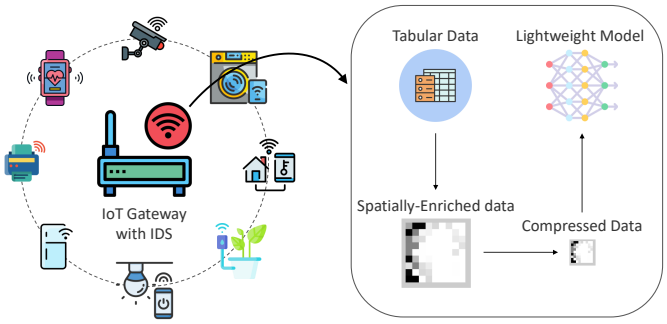

Fig. 1: Illustration of an intrusion detection system in IoT compact devices where the requirements on real-time traffic processing are often challenging to satisfy, particularly if state-of-the-art (SOTA) computer vision techniques are used.

Recently, there has been significant emphasis on integrating deep learning (DL) techniques to enhance intelligence in IDSs, particularly for IoT networks or near the source (e.g., at an IoT gateway [4] or software-defined networking controller [5], [6], as shown in Figure 1). However, the limited computing power and memory size in IoT systems present challenges for incorporating DL methods, which often have large model sizes. Researchers are becoming more interested in creating lightweight IDSs with real-time detection capability in IoT networks [7], [8]. The strategies for enhancing the efficiency and lightweight capability of deep learning methods often focus on two primary factors: 1) reducing the dimensionality of traffic features, and 2) decreasing the learning model complexity. Both of these approaches aim to reduce computing complexity. The first method achieves this indirectly by reducing network traffic feature dimensionality [9], [10], while the second method directly lessens the complexity of the DL learning layer structure, making the proposed approach more

V. L. Nguyen and H. P. Tsai are with the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan (e-mail: nvlinh@cs.ccu.edu.tw; haoping0901@gmail.com).

V. L. Nguyen is also with the Advanced Institute of Manufacturing with High-tech Innovations, National Chung Cheng University, Chiayi, Taiwan.

H. D. Shin is with Kyung Hee University, South Korea (e-mail: hshin@khu.ac.kr).

T. Q. Duong is with Memorial University, Canada and Queen's University Belfast, UK (e-mail: tduong@mun.ca).

This work was supported in part by the National Science and Technology Council (NSTC) of Taiwan under Grants 111-2222-E-194-007-MY2, 112-2221-E-194-017-MY3, and in part by the Advanced Institute of Manufacturing with High-tech Innovations (AIM-HI) from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan. The work of H. Shin was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (NRF-2022R1A4A3033401). The work of T. Q. Duong was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109.

Corresponding author: Van-Linh Nguyen.

lightweight [11]. While many researchers have successfully reduced model complexity and achieved promising performance through these two approaches, striking a good balance between the two approaches remains an open research issue.

In order to tackle the identified concerns, we develop a novel NIDS by exploiting the inherent characteristics of vision transformer-based techniques and spatial transformation data of network traffic. Accordingly, we introduce a transformer-based lightweight architecture that effectively balances model complexity and performance. The IDS can reduce the feature dimension by compressing the data generated by the Image Generator for Tabular Data (IGTD) technique [12]. The study's contributions are as follows.

1) The first attempt to propose a lightweight vision transformer(ViT)-based learning model for malicious traffic detection, leveraging the inherent characteristics of computer vision-based techniques. ViT uses spatial relationships in network traffic and multihead self-attention mechanisms to focus on crucial information and generalize from training data to identify and flag novel or zero-day attacks with fewer learning layers.
2) The first attempt to exploit IGTD transformation techniques and data compression to enhance learning efficiency. Accordingly, this computer vision-based feature dimensionality reduction does not affect the learning capability. The data compression approach not only decreases the feature dimension but also preserves the most crucial features, contributing to an effective balance between data efficiency and information retention.
3) The system is evaluated on the benchmark public dataset, CICDDoS2019 [13]. Experimental results indicate that the proposed system maintains both outstanding classification accuracy and significantly lower model complexity and model size compared to state-of-the-art (SOTA) studies. Moreover, experiment results show that the model complexity and system performance are resistant to changes in input size.

The remaining sections of this paper are organized as follows. Section II provides an overview of existing research related to IDS. Section III presents the system model and problem statement. Section IV offers detailed explanations of the components of the proposed framework. Section V outlines the data preprocessing methods as well as experimental results. Finally, Section VI concludes the work.

II. Related Work

Recently, DL approaches with network traffic feature learning capabilities have become popular for DDoS mitigation [14], [15]. For example, the studies in [16] and [17] proposed a time-based extraction method based on one-dimensional convolutional neural network (CNN) for DDoS detection. Experimental results demonstrate superior performance compared to a variety of DL methods, both in simulated environmental data and the CICDDoS2019 dataset. The authors in [18] introduced an intrusion detection model consisting of two cascaded detection tiers. The system integrates Recurrent Neural Networks (RNN) with an enhanced backpropagation algorithm, enabling commendable performance on both balanced and imbalanced class data from the NSL-KDD dataset. The study [4] used a transformer and CNN to identify DDoS attacks. The recommended solution outperforms the latest deep learning DDoS intrusion detection algorithms on the CICDDoS2019 dataset. Gupta et al. [19] developed LID-IDS, a two-layer IDS, to detect network intrusions. The first layer of LID-IDS uses LSTM binary classifier to detect attack traffic. An ensemble approach with enhanced one-on-one handles frequent and infrequent network incursions in the second layer. The same approach is proposed in the studies [20], [21] but exploits depthwise separable convolution and bidirectional long short-term memory technique. The authors in [22] suggested a hybrid model consisting of three blocks. The first block reconstructs features using 1D CNN with a residual link. The second block extracts features using LSTM and GRU. CNN forms the third block for cyberattack detection.

While numerous methods have demonstrated high performance in intrusion detection, the practicality of deploying models with high complexity to devices with limited computing resources is sometimes overlooked. Therefore, the efficacy of feature extraction techniques becomes a crucial factor in advancing DDoS defense methods [29]. This is particularly relevant because manually analyzing complete raw network traffic samples within extensive feature sets is both impractical and cost-prohibitive, especially when not all features offer pertinent insights for detecting malicious payloads. For example, Sanchez et al. [23] conducted a feature importance analysis using the Analysis of Variance (ANOVA) statistical method to detect DDoS traffic. Wei et al. [24] introduced AE-MLP, a hybrid model integrating autoencoder (AE) and multi-layer perception (MLP) for precise DDoS attack classification by extracting vital features from raw network traffic data. The authors in [30] employed a long short-term memory (LSTM) autoencoder to effectively identify abnormal traffic through feature dimensionality reduction, a promising intrusion detection approach. Federated learning-based approach [31] is an emerging technique but is complicated in management.

In addition to the aforementioned methods that aim to enhance efficiency by reducing feature dimensionality, several researchers have gone further by incorporating models with lower complexity [32]. This approach allows the IDS engines to achieve a certain level of accuracy with reduced model complexity. Table I summarizes the review of related work for DL-based approaches and the proposed method's contributions, with specific measurements of the lightweight (e.g., model size, memory computation). For example, the authors in [25] introduced Lucid, a lightweight DDoS detection system, by leveraging a limited set of specific features. Although their proposed approach demonstrated promising results on various datasets, the balance for accuracy and memory computation requirement is still low [33]. The authors in [26] proposed an innovative network intrusion detection approach for IoT utilizing a lightweight deep neural network (LNN). Their approach incorporates principal component analysis (PCA) to decrease dimensionality and employs various classifier techniques, including expansion and compression structures, inverse residual structures, and channel shuffle operations, to

TABLE I: Summary of several lightweight DL-based related studies and novelty contributions (performance|ranking)

Reference	Feature selector	Learning model	Accuracy (%|#)	Memory computation (FLOPS|#)	Model size (KB|#)	Response time (ms|#)
ANOVA-DNN [23]	ANOVA	MLP	97.51|7	7,298|4	40.72|5	2090|4
AE-MLP [24]	AutoEncoder	MLP	98.53|6	3,957|3	24.64|2	2004|1
Lucid [25]	-	1DCNN	98.83|5	3,266|2	39.20|4	2039|2
LNN [26]	PCA	LNN	99.31|4	111,458|6	304.03|6	2535|6
CNN-LSTM [27]	XGB	CNN+LSTM	99.62|1	36,028|5	29.55|3	2580|7
DCNN [28]	PCA	DCNN	99.45|3	167,618|7	311.32|7	2510|5
Ours	Compressed Data from IGTD	ViT-based Lightweight Model	99.57|2	**2,322|1**	**13.80|1**	2068|3

TABLE II: Summary of the notations used in this study.

Notation	Meaning
N	Number of features
k	Side length of the compressed feature
$r_{i,\ j}$	The value assigned to the i-th and j-th feature pair
$q_{i,\ j}$	The value assigned to the i-th and j-th pixel coordinate pair
$coor_i$	The coordinate of the i-th pixel
(H, W)	The resolution of the original image
N_p	Number of patches
(P, P)	The resolution of each image patch
C	The number of channels in the input image
D	The constant size of latent vector in transformer
h	The frequency of the self-attention mechanism being applied.
D_h	The size of the vector transformed before entering the self-attention mechanism
L	Number of stacked transformer encoders

extract features effectively while maintaining low computational expenses. Zainudin et al. [27] employ ML-based IDS approaches and a hybrid architecture consisting of CNN and LSTM, with the CNN in this architecture utilizing factorization to achieve low model complexity. The study in [34] and [32] introduced a novel vision transformer-based framework for IDS. However, none of these studies target a good balance in accuracy, memory computation, response time, and a lightweight architecture for low-cost IoT gateways.

Despite various considerations of lightweight approaches, these methods either employ a considerable number of learnable parameters to enhance accuracy or sacrifice accuracy to maintain lower model complexity. Unlike prior work, this article presents a novel lightweight IDS method for DDoS detection, striking a better trade-off between model complexity and accuracy. Table II outlines the main notations used in this article and their meanings.

III. Attack model and problem formulation

This section details the attack model and DDoS datasets, followed by the problem formulation.

A. Attack model

This work assumes that the attacker initiates various types of DDoS attacks using compromised IoT devices. The primary objective is to deplete the target server's resources. For evaluation, the well-known dataset CICDDoS2019 is used [13]. Bot-IoT [35] and DoS/DDoS-MQTT-IoT [36] with DoS attack traffic from IoT networks using the Message Queueing Telemetry Protocol (MQTT), such as refrigerators, smart garage doors, weather monitoring, smart lights, and smart thermostats, can be utilized to calculate attack detectors' response time. Nevertheless, as Bot-IoT and DoS/DDoS-MQTT-IoT are not often used in state-of-the-art (SOTA) research, in this article, the well-recognized CICDDoS2019 dataset is still used for major performance evaluation comparison scenarios.

B. Problem statement

This study considers classifying the DDoS attack traffic into two types: abnormal traffic and legitimate traffic. The objective is to gain information about a function $f : X \rightarrow Y$, where $Y < \infty$. Suppose that a DL algorithm denoted as A runs on a dataset $D = \{d_1, \ldots, d_i, \ldots, d_n\}$ consisting of training data points. Each data point $d_i = \{x_1, \ldots,\ x_i, \ldots,\ x_N\}$ represents a network flow characterized by N features, e.g., flow duration, total packets in the forward and backward directions, packet size, number of flow packets, protocols, attack states, and additional attributes, as in CICDDoS2019 dataset guidelines.

In order to simplify the DL model, we employ a spatially enriched data transformation method to convert each data point x_i from its original space $\mathbb{R}^{1\times N}$ to $\mathbb{R}^{\lceil \log_2 N\rceil \times \lceil \log_2 N\rceil}$, and subsequently compress it to a space with fewer dimensions $\mathbb{R}^{k\times k}$, where k is a positive integer less than or equal to $\lceil \log_2 N\rceil$. The reduced feature vectors then serve as input data for a lightweight deep-learning algorithm. The reduced feature vectors enable the proposed method to train faster and thus help to enhance the detection speed.

IV. Proposed Data Transformation and Enhanced Intrusion Detection Method

This section introduces a novel data transformation to enhance intrusion detection systems by enabling DDoS tabular data transfer into computer vision data and utilizing a state-of-the-art computer vision technique for feature extraction and DDoS attack traffic classification. The proposed IDS architecture is illustrated in Figure 2. As illustrated in the right

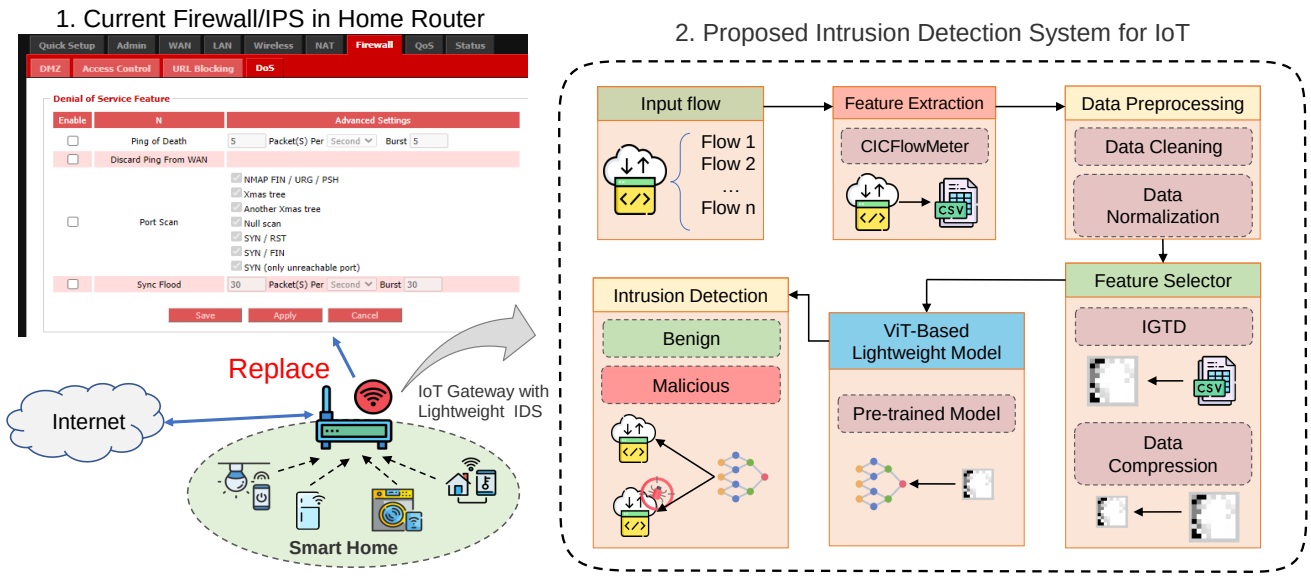

Fig. 2: The architecture of the proposed intrusion detection system for IoT networks and the details of data transformation.

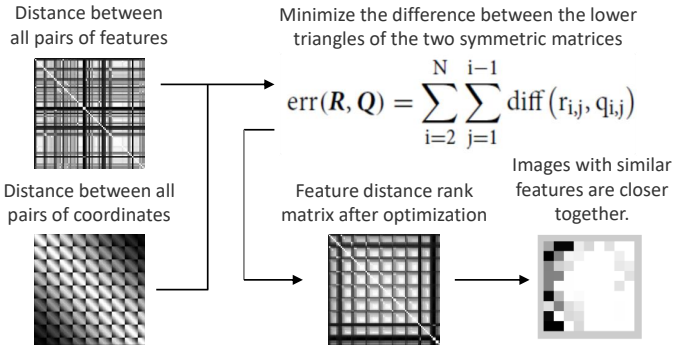

Fig. 3: The workflow of the proposed data transformation algorithm.

side of Figure 2, the architecture comprises several processing steps. Initially, CICFlowMeter-V3 [37] extracts features from collected traffic. To ensure data quality, the retrieved features are preprocessed, which includes cleaning and normalization. Subsequently, by compressing the transformed data obtained from the spatial data transformation step, the system can reduce feature dimensions. This strategy enables the proposed vision transformer-based learning model to reduce model complexity while retaining good performance in IoT gateways (e.g., replacing current IDS of home routers as in Figure 2).

A. *Spatial data transformation mechanism*

Most tabular data (e.g., network traffic saved in CSV or pcap files) lacks an inherent spatial connection, making it unsuitable for computer vision-based methods. To alleviate this restriction, the following data transformation approach was devised. The purpose of this approach is to turn tabular data into pictures and arrange comparable characteristics near together inside the image by giving them precise pixel coordinates.

The complete process of the data transformation approach is shown in Figure 3. Initially, the data transformer uses the Euclidean distance to compute pairwise feature distances. These values are then sorted in ascending order, and values are allocated depending on the sorted sequence, with smaller values awarded to pairs of features with shorter distances and bigger values to those with longer distances. These given values are recorded in a matrix R with dimensions N by N, where N denotes the number of features. In matrix R, the element $r_{i,j}$ in the i-th row and j-th column indicates the distance between the i-th feature and j-th feature over all the sample data in set S. $r_{i,j}$ is expressed by

$$r_{i,j} = \sqrt{\sum_{k=0}^{S}(x_{i_k} - x_{j_k})^2} \tag{1}$$

Notably, the distance values between the i-th and j-th features are symmetric, meaning that $r_{i,j} = r_{j,i}$, resulting in R being essentially a diagonal matrix.

Additionally, the data transformer computes the distance between every pair of pixel coordinates, also employing the Euclidean distance as the metric. These distances are sorted, with smaller values assigned to pairs of pixels with closer distances and larger values to those with greater distances. The assigned values are stored in a matrix Q of size N by N, where N denotes the number of pixel coordinates. In matrix Q, the value $q_{i,j}$ in the i-th row and j-th column denotes the distance between the i-th and j-th pixel coordinates as

$$q_{i,j} = \sqrt{(coor_i - coor_j)^2} \tag{2}$$

Pixel coordinate distances, like matrix R, are symmetric, which means that $q_{i,j} = q_{j,i}$. Thus, matrix Q is effectively a diagonal matrix. With these two matrices, the data transformer proceeds to cluster similar features together and position dissimilar features farther apart in terms of pixels. This is accomplished by minimizing an error function $err(R, Q)$. $err(R, Q)$ is expressed by the difference between $r_{i,j}$ and $q_{i,j}$

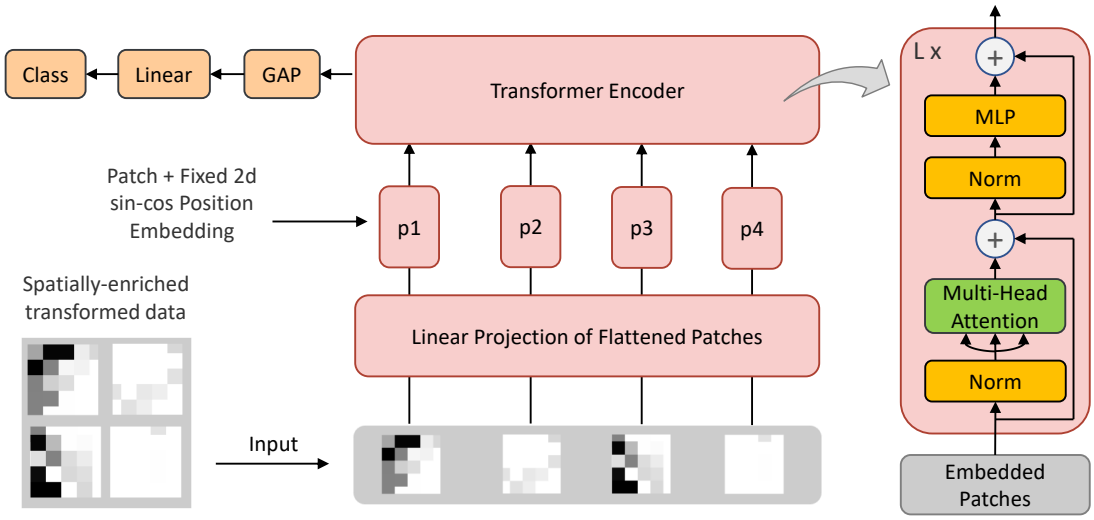

Fig. 4: Overview of the proposed architecture for DDoS classification on transformed data.

as follows:

$$err(R,Q) = \sum_{i=2}^{N} \sum_{j=1}^{i-1} |r_{i,j} - q_{i,j}| \tag{3}$$

Essentially, this process calculates the disparity between the distances of features and the distances of their positions. The process of minimizing the error function involves clustering similar features together and increasing the distance between different features. The error function is minimized by reordering the rows and columns of the R matrix. After minimizing the error function, the updated R matrix represents the ordered arrangement of data with spatial information. The data transformer then sequentially places the features into the image according to this order, thereby creating transformed data with a spatial relationship.

B. Proposed vision transformer-based learning for DDoS classification

After Dosovitskiy et al. [38] introduced ViT, self-attention-based architectures finally gained the capability to compete with the established supremacy of CNN-based architectures in handling computer vision tasks. At the inception of self-attention-based architectures, especially Transformers [39], they became the favored models in natural language processing. Nevertheless, CNNs have continued to dominate computer vision-related techniques. The need for enhancement in self-attention-based designs stems from the intrinsic inductive biases of CNNs, including translation equivariance. ViT tackles this difficulty by training on bigger datasets, allowing it to overcome inductive biases and maximize its ability to attend to global information, resulting in enhanced classification results.

In this study, we have identified a unique feature of ViT-based architectures (i.e., discriminative learning, which is often seen in image classification of computer vision) to effectively address the DDoS categorization challenge. Further, we strategically adopt an enhanced version of ViT [40] to improve the learning. This variant, compared to the original version, incorporates adjustments that simplify its architecture while significantly enhancing the performance of ViT. The self-attention mechanisms in ViT can capture global dependencies to identify anomalies and generalize from training data that help the model recognize and flag novel or zero-day attacks. Figure 4 illustrates the fundamental overview of the proposed method's architecture for DDoS classification. At first, the two-dimensional images x with dimensions $H \times W \times C$ are transformed into patches x_p with dimensions $N_p \times (P^2 \cdot C)$. Let (H, W) be the dimensions of the original picture, C represent the number of channels, (P, P) represent the dimensions of each image patch, and N_p be the resultant number of patches. N_p is computed as $\frac{HW}{P^2}$. Subsequently, these patches are subjected to an embedding procedure using a trainable linear mapping with a fixed vector size D, which is defined by the hyperparameter dim. These embedded patches are then updated using position embeddings E_{pos} to generate integrated patches z_0, which serve as the input to the transformer encoder. z_0 is calculated by

$$z_0 = [x_p^1 E; x_p^2 E; \ldots; x_p^{N_p} E] + E_{pos} \qquad E \in \mathbb{R}^{(P^2 \cdot C) \times D}$$
$$E_{pos} \in \mathbb{R}^{N_p \times D} \tag{4}$$

The transformer encoder consists of two components: a multihead self-attention (MSA) component and an MLP component. Layer normalization is performed first, followed by the application of residual connections. The MSA block employs the self-attention (SA) mechanism on the input h times, as seen on the right side of Figure 5. The value of h may be adjusted by the hyperparameter heads. Before entering the self-attention mechanism, the input is converted into three vectors, Q, K, and V, each with a size of D_h, which will serve as the following inputs:

$$[Q, K, V] = zW^{qkv} \qquad W^{qkv} \in \mathbb{R}^{D \times 3D_h} \tag{5}$$

The transformed vectors pass the scaled dot-product attention process (shown on the left of Figure 5), resulting in the weights matrix O_{sdpa}. This matrix is obtained by computing the dot

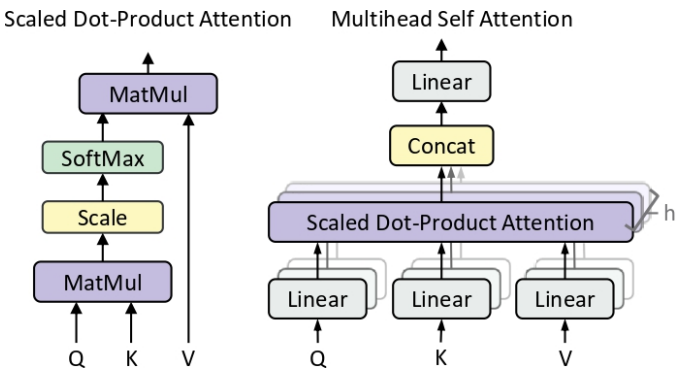

Fig. 5: The graphic illustrates the structure of multihead self-attention. The diagram's left side showcases the Scaled Dot-Product Attention mechanism, while the right side demonstrates the multihead self-attention mechanism [39].

products of Q and K, scaling the result by $\sqrt{D_h}$, applying a softmax function, and finally utilized to obtain weights on V as by

$$O_{sdpa} = softmax(\frac{QK^T}{\sqrt{D_h}}) \tag{6}$$

$$SA(z) = O_{sdpa}V \tag{7}$$

The self-attention process produces h outputs in parallel, which are subsequently concatenated and linearly transformed to form the final result by

$$MSA(z) = [SA_1(z), \ldots, SA_h(z)]W^O \quad W^O \in \mathbb{R}^{h \cdot D_h \times D} \tag{8}$$

The output of the l-th MSA block, where $l \leq L$ and L represents the number of stacked encoders controlled by the hyperparameter depth in the transformer encoder, denoted as z'_l, incorporates layer normalization and residual connections. z'_l is expressed by

$$z'_l = MSA(LN(z_{l-1})) + z_{l-1}, \qquad l = 1 \ldots L \tag{9}$$

This output is used as the input for the MLP block. The MLP block is made up of two linear transformation layers with GELU non-linearity. The hyperparameter mlp_dim controls the dimensionality of the inner layer. Furthermore, the MLP block begins with layer normalization and is followed by the addition of residual connections to the output z_l by

$$z_l = MLP(LN(z'_l)) + z'_l, \qquad l = 1 \ldots L \tag{10}$$

The output of the MLP block is then transmitted via a global average pooling (GAP) layer and a layer normalization layer. Finally, a linear layer is used to get the classification result by

$$y = LN(GAP(z_L)) \tag{11}$$

C. Proposed low complexity learning model for IoT devices

In light of the preceding introduction, several key insights can be summarized as follows.

1) Data transformer proves its proficiency in aggregating akin features within close proximity within an image.

TABLE III: Configuration of the SimpleViT architecture.

Hyperparameters	Value
image_size	7×7
patch_size	7×7
dim	8
depth	1
heads	1
mlp_dim	8
channels	1

Fig. 6: The device on which we conducted experiments.

2) Vision Transformer has the ability to collect global information, but its efficacy is assured only when a sufficient number of encoders are used.

Building upon these two fundamental observations, the data transformer plays a key role in generating spatially correlated data. This process entails the combination of similar features while segregating dissimilar ones. Furthermore, as a consequence of aggregating similar data, the direct resizing of the generated images can indirectly achieve the objectives of feature extraction and data compression. Subsequently, we implement an update of the Simple Vision Transformer (SimpleViT) architecture by utilizing only a single encoder block for DDoS attack classification. While SimpleViT's typical ability to access global information relies on the presence of a sufficient number of encoders, the feature aggregation in the proposed method helps to advance high performance with a reduced number of encoders. The detailed hyperparameter configuration for the revised architecture can be found in Table III. The cross-entropy loss function is used in the training. The other settings are the optimizer (Adam), learning rate (0.001), and training batch size (64).

V. Evaluation & Discussion

This section provides a comprehensive analysis of the performance of the proposed system while compressing data generated by the spatially enhanced data transformation technique at various sizes. In addition, we validate the proposed learning strategy by enhancing the model's learning capabilities through the adjustment of several hyperparameters. Moreover, the proposed method's performance is also compared with that of SOTA studies in order to validate their advantages,

particularly the capability to balance the trade-off between complexity and detection accuracy. Evaluations are conducted on the Nvidia Jetson Nano, which serves as an IoT gateway or home router. The device is equipped with a Cortex-A57 CPU and 4GB of RAM, as seen in Figure 6.

A. Dataset description and data preprocessing

For comparison purposes with related SOTA studies, the evaluation was conducted utilizing the publicly available CICDDoS2019 dataset [13], curated by the Canadian Institute for Cybersecurity at the University of New Brunswick, focusing on DDoS attack traffic. This dataset comprises both benign traffic and common DDoS attacks that are divided into two distinct segments: the training day and the testing day on real-world network scenarios. The collection comprises CICFlowMeter-V3 network traffic analysis results and PCAP files [37]. The analysis yields labeled flows containing various features and attack types. According to [13], the dataset includes 13 DDoS attack traffic on several typical protocols: lightweight directory access protocol (LDAP), simple network management protocol (SNMP), simple service discovery protocol (SSDP), UDP-Lag, or PortScan. As depicted in Figure 7, the attacks are categorized into two types:

1) *Reflection-Based DDoS Attacks*: Attackers use reflection-based attacks by substituting the source IP with the victim's IP and flooding fraudulent requests to legitimate services (known as reflectors). As a result, the victim's IP address is flooded with reply packets produced by the servers while it is difficult to track back the attacker's identity. The term "reflection" stems from the utilization of the same TCP/UDP protocol in both directions but takes advantage of legitimate servers and IP spoofing to take down the target. As shown in Figure 7, MSSQL and SSDP fall under TCP-based attacks, while NTP and TFTP are categorized as UDP-based attacks. Other types of DDoS attacks, such as DNS, LDAP, NetBIOS, SNMP, and PORTMAP, employ techniques that utilize both TCP and UDP protocols.
2) *Exploitation-Based DDoS Attacks*: This kind of attack takes advantage of the operational procedures of several protocols, such as TCP-based SYN flooding, UDP-based UDP flooding, and UDP-Lag attacks. These attacks result in the victim server allocating resources to incomplete connection attempts originating from the faked IP addresses, leading to resource depletion and rendering the victim server unreachable.

Regarding the dataset structure, the training day encompasses 12 distinct types of DDoS traffic, whereas the testing day incorporates 7 such traffic categories. As summarized in Table IV, labels are present in the training and testing datasets. We randomly sampled in 30,000 records for each attack category on the training day and 10,000 records for each attack category on the testing day while retaining all benign ones from both the training and testing day.

To ensure the dataset's integrity, several preprocessing steps are required before conducting experiments as follows.

TABLE IV: The quantity of sampled instances per class within the subset of the CICDDoS2019 dataset, as referenced in [41].

Label	Training records	Testing records
LDAP	30,000	10,000
MSSQL	30,000	10,000
NetBIOS	30,000	10,000
UDP	30,000	10,000
SYN	30,000	10,000
UDPLag	30,000	10,000
Benign	53,796	51,241

1) *Data cleaning*: The original dataset contains several non-contributory features. Therefore, we exclude ten of these features, such as "Flow ID", "Fwd Header Length.1", "Destination IP", "Source IP", "Source Port", "Destination Port", "Timestamp", "Unnamed: 0", "Inbound", and "SimillarHTTP". Additionally, we encode the labels such that benign instances are assigned a value of 0, and malicious instances are assigned a value of 1, similar to handling a binary classification task. In addition, any missing values, occurrences with infinity, and duplicate values are removed, resulting in a final set of 77 features for assessment.
2) *Data normalization*: The CICDDoS2019 dataset includes features that exhibit substantial variation in values, ranging from the lowest to the highest values. These features include "Flow Duration", "Flow IAT Std", "Flow IAT Max", and "Bwd IAT min". Data normalization may effectively mitigate the significant variability seen in these characteristics. Data normalization may decrease the time it takes to train a model, speed up convergence, and lessen the likelihood of gradient explosion. In this work, a min–max normalization procedure, as described in Eq. 12, is used to normalize the data. Accordingly, normalized data X_{norm} is expressed by

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}, \tag{12}$$

where the variable X_{norm} indicates a normalized numeric value that ranges from 0 to 1. The variable X_{max} represents the greatest value of the feature, while X_{min} represents its lowest value.

B. Measurement metrics and evaluation

This research employs four measures to assess the performance of the proposed system: accuracy, precision, recall, and F1-score. These measures are based on the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Accuracy represents the proportion of correct predictions made by the model out of the total number of predictions. Precision specifically focuses on the accuracy of positive predictions. Recall is concerned with finding all real positive situations. The F1-score gives a thorough evaluation by balancing precision and recall. Here are the expressions for the metrics.

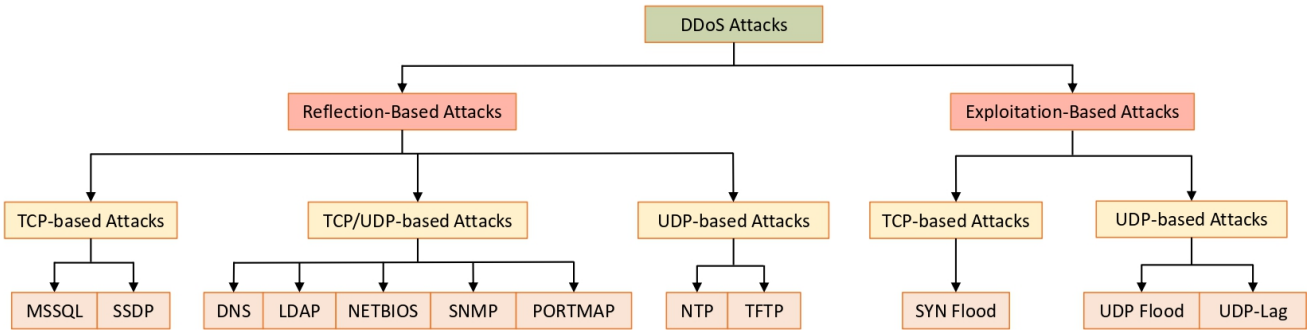

Fig. 7: DDoS attack types and the corresponding protocols in CICDDoS2019 [13].

TABLE V: Proposed model performance in terms of complexity and model size with various input and patch sizes.

Image size	Patch size	Accuracy (%)	F1-score (%)	Learnable Parameters	FLOPs	Model Size (KB)
3×3	3×3	99.10	99.21	2,372	2,372	**11.72**
4×4	2×2	99.55	99.63	**2,322**	**2,322**	17.56
5×5	5×5	99.44	99.53	2,532	2,532	12.56
6×6	3×3	99.36	99.45	2,372	2,372	18.00
7×7	7×7	**99.57**	**99.65**	2,772	2,772	13.80
8×8	4×4	99.54	99.62	2,442	2,442	18.62
9×9	3×3	99.50	99.63	2,372	2,372	28.46

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

$$Precision = \frac{TP}{TP + FP}; \quad Recall = \frac{TP}{TP + FN}$$

$$F1 - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

C. The effect of various degrees of dimensionality deduction

Reducing input dimensionality is important in this approach. The first scenario tests performance across several dimensionality reduction levels to demonstrate the usefulness of the proposed technique for two performance metrics. To use created spatial linkages and balance model complexity and learnable parameters, the patch size is used. Table V presents the model complexity and patch size. Evaluation results reveal that the highest accuracy reached is 99.57%, with an F1-score of 99.65%, when utilizing an image size and patch size of 7×7 and 7×7, respectively. Conversely, when configuring the image size as 4×4 and the patch size as 2×2, we observe a slight decrease in performance, accompanied by the lowest number of learnable parameters. However, this configuration creates more patches, making the model bigger than models with greater input sizes and fewer patches. Generally, setting the patch size to the same dimensions can maintain the amount of learnable parameters. Still, the model's performance improves with an increase in input size. This is because in the proposed design, patch size is the only element affecting the number of learnable parameters during embedding. Note that the embedding process represents only a small portion of the architecture. Essentially, the approach intuitively demonstrates simplicity that helps to reduce the memory computation and model size of the proposed method. This feature makes it especially feasible for implementing IoT gateways or devices that have limited resources.

D. The effect of hyperparameter configurations

The proposed scheme is also evaluated with multiple hyperparameters to demonstrate the spatial data transformation mechanism's strength. Figure 8 shows the impact of varying the value of the hyperparameter depth, which controls the number of learnable parameters and model performance. It is evident that increasing the number of stacked transformer encoders leads to a minor improvement in performance ($0.05\% = 99.62\% - 99.57\%$), the number of learnable parameters increases by more than fivefold (from 2,722 to 13,892). Figure 9 presents the effect of adjusting the hyperparameter dim/mlp_dim, which controls the transformer MLP output size and the dimensionality of the MLP inner layer. It is noticeable that increasing the MLP output size may even lead to a decrease in the detection accuracy performance. For instance, when the hyperparameter dim and mlp_dim are set to 64, the number of learnable parameters increases by over tenfold (from 2,722 to 28,644), but the detection accuracy performance decreases by $0.01\% = 99.57\% - 99.56\%$.

On the other hand, although setting hyperparameters dim and mlp_dim decrease to 32 results in a slight improvement of $0.12\% = 99.69\% - 99.57\%$, the number of learnable characteristics increases by more than four times, from 2,722 to 12,324. Considering achieving a better balance between model complexity and performance, sacrificing a slight accuracy performance improvement for fewer model parameters seems to be a preferable choice. Figure 10 illustrates the

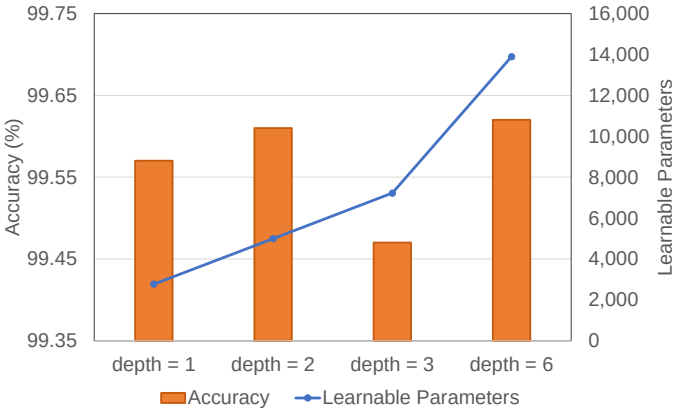

Fig. 8: The impact of varying configurations of the depth hyperparameter on accuracy and the number of learnable parameters.

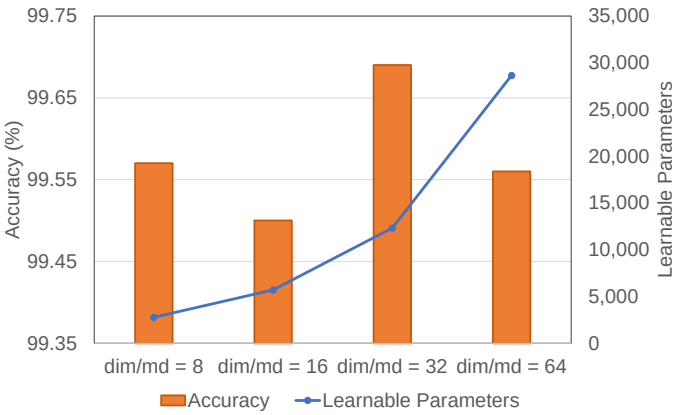

Fig. 9: The impact of varying configurations of the dim/mlp_dim hyperparameter on accuracy and the number of learnable parameters.

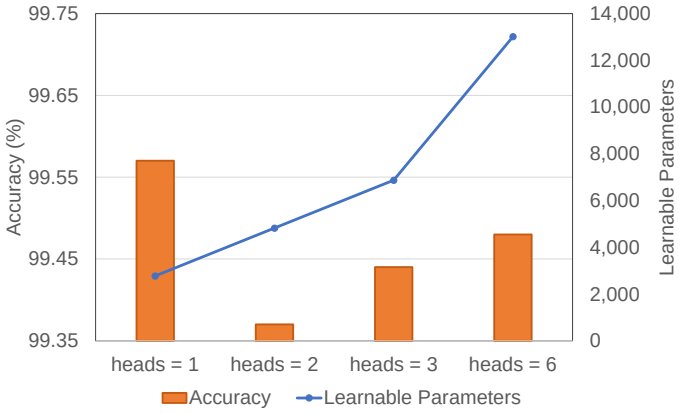

Fig. 10: The impact of configurations of the heads hyperparameter on accuracy and the number of learnable parameters.

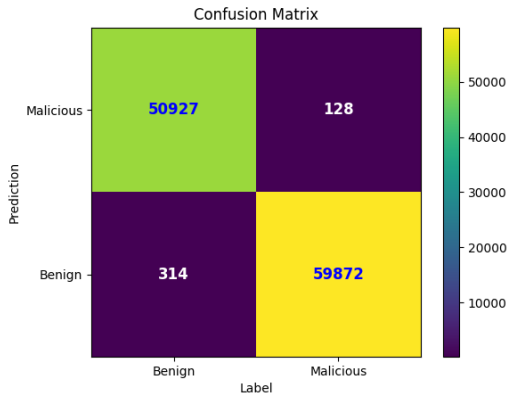

Fig. 11: Image size set to 4×4.

impact of varying the hyperparameter heads, which determines the frequency of attention mechanism execution in the transformer, on the number of learnable parameters and model performance. Increasing the number of hyperparameter heads may not improve model accuracy performance, yet it significantly increases the number of learnable parameters by over fourfold (from 2,722 to 13,012). As a result, the results indicate that the proposed spatial data transformation mechanism effectively retains model accuracy performance while reducing the number of learnable parameters in the ViT-based detection architecture.

E. Performance comparison with the related SOTA studies

Table VI and Table VII show the results for comparison between the proposed system's performance and that of existing studies using the same dataset and input size. The evaluation results indicate that the proposed approach's performance is very competitive in terms of model complexity and size. Accordingly, when the input size is set to 4×4 as shown in Figure 11, the proposed method's accuracy and F1-score are slightly lower than the CNN-LSTM architecture. However, the system outperforms the CNN-LSTM architecture regarding model complexity. As shown in Table VI, the proposed method achieves a remarkable $55.35\% = (1 - \frac{2322}{5200}) \times 100\%$ reduction in learnable parameters and an impressive 93.56% reduction in FLOPs.

In contrast, other methods, such as Lucid, ANOVA-DNN, and AE-MLP, despite having architectures with a limited number of learnable parameters and FLOPs, fail to deliver acceptable performance on the same dataset. Furthermore, while the proposed method's model size is slightly larger than that of the Lucid and AE-MLP architectures, the model size increase is only 3.84 KB and 0.58 KB, respectively. This small increase in model size results in an accuracy improvement of $0.72\% = 99.55\% - 98.83\%$ and $1.02\% = 99.55\% - 98.53\%$, respectively. Regarding reaction time, both the LNN and CNN-LSTM designs have greater FLOPs than other approaches, resulting in longer response times. Although the AE-MLP design has the fastest response time, the proposed approach works effectively with just a tiny delay (0.2ms).

If the input size increases to 7×7, as shown in Figure 12, this work continues to demonstrate competitive performance, with accuracy and F1-score only slightly lower than the LNN architecture. As results shown in Table VII, the proposed method achieves a remarkable $82.07\% = (1 - \frac{2,772}{15,458}) \times 100\%$ reduction in learnable parameters and an impressive $99.28\% =$

TABLE VI: The proposed method's performance in comparison with that of other SOTA methods in terms of model performance, complexity, size, and response time using the same dataset and input size of 4×4.

Algorithms	ANOVA-DNN [23]	AE-MLP [24] (only MLP part)	Lucid [25]	LNN [26]	CNN-LSTM [27]	DCNN [28]	Proposed Model
Accuracy (%)	97.51	98.53	98.83	99.31	**99.62**	99.45	99.55
Precision (%)	96.60	98.12	98.52	99.09	**99.58**	99.29	99.48
Recall (%)	98.95	99.27	99.41	99.72	**99.82**	99.79	99.79
F1-score (%)	97.76	98.69	98.96	99.40	**99.70**	99.54	99.63
Learnable Parameters	7,298	3,957 (with encoder)	3,266	15,458	5,200	20,738	**2,322**
FLOPs	7,298	3,957 (with encoder)	3,266	111,458	36,028	167,618	**2,322**
Model Size (KB)	30.55	16.98 (with encoder)	**13.72**	132.63	29.55	159.13	17.56
Response Time (ms)	2.090	**1.949**	2.039	2.535	2.580	2.510	2.188

TABLE VII: The proposed method's performance in comparison with that of other SOTA methods in terms of model performance, complexity, size, and response time using the same dataset and input size of 7×7.

Algorithms	ANOVA-DNN [23]	AE-MLP [24] (only MLP part)	Lucid [25]	LNN [26]	CNN-LSTM [27]	DCNN [28]	Proposed Model
Accuracy (%)	98.78	99.08	99.06	**99.61**	99.52	99.44	99.57
Precision (%)	98.41	98.97	99.12	99.51	99.36	99.22	**99.55**
Recall (%)	99.44	99.42	99.23	**99.86**	99.84	99.83	99.75
F1-score (%)	98.92	99.20	99.18	**99.69**	99.60	99.53	99.65
Learnable Parameters	9,806	5,805 (with encoder)	9,602	15,458	5,600	20,738	**2,722**
FLOPs	9,806	5,805 (with encoder)	9,602	385,634	109,308	490,754	**2,722**
Model Size (KB)	40.72	24.64 (with encoder)	39.20	304.03	48.96	311.32	**13.80**
Response Time (ms)	2.146	**2.004**	2.127	2.718	2.722	2.640	2.068

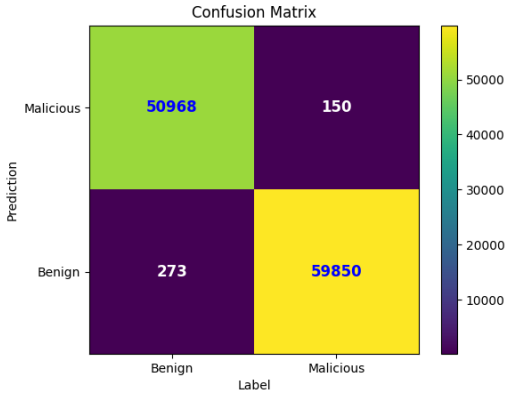

Fig. 12: Image size set to 7×7.

$(1 - \frac{2,772}{385,634}) \times 100\%$ reduction in FLOPs. When the input size increases, other methods tend to exhibit an increase in either the number of parameters or FLOPs. Meanwhile, if the input size grows, the proposed approach reduces learnable parameters and FLOPs significantly.

Regarding memory capacity, it is evident that the proposed approach reduces the amount of storage required if the input size increases from 4×4 to 7×7. This is because when both the input size and patch size are set to 7×7, fewer patches are generated, resulting in the proposed architecture occupying less space in both forward and backward processes. In terms of response time, the AE-MLP architecture stands out with a minimum response time of 2.004ms among all methods. This work follows closely with a slightly shortened response time while maintaining high performance. In conclusion, this system is great for deployment on IoT devices with limited resources since it can find an acceptable balance between performance, learnable parameters, and FLOPs.

F. Discussion on the worst case and the best case of performance for reproducibility

In this subsection, we explore the best and worst-case scenarios of performance and identify strategies to setup at best, providing readers with a comprehensive understanding to reproduce our work. The best-case scenario occurs when the hyperparameters dim/mlp_dim are set to 32, as depicted in Figure 9. This is because moderately increasing these hyperparameters allows the model to learn more complex patterns. However, excessive increments in these hyperparameters may lead to overfitting issues (as evidenced by the performance drop when hyperparameters dim/mlp_dim are set to 64 compared to 32 in Figure 9). Therefore, when there are sufficient computational resources available for deployment

on IoT devices, increasing the hyperparameters dim/mlp_dim should be done within a reasonable range.

The worst-case scenario occurs when the hyperparameter heads are set to 2, and increasing the number of heads further still results in performance degradation compared to before, as illustrated in Figure 10. This is because increasing this hyperparameter enhances the model's ability to learn different patterns. However, our employed spatial data transformation mechanism effectively aggregates important information, allowing the model to achieve high performance even when learning a single pattern. Increasing the hyperparameter heads, on the other hand, causes the model to focus on less important patterns, leading to the need for more learnable parameters (e.g., increasing hyperparameters such as depth, dim, and mlp_dim) to prevent the model from being confused by less important information. Therefore, even if there are sufficient computational resources available, this parameter should be kept as small as possible to enable the model to focus on the regions where important information is aggregated.

G. Generalization and adaptivity performance on unseen data

To quantify the generalization and adaptivity of the proposed method when applied to other datasets not seen during the training phase, we have conducted additional experiments to evaluate the proposed model on different IoT datasets such as Bot-IoT [35] and DoS/DDoS-MQTT-IoT [36]. The results show that the model retains its effectiveness, with only a slight decrease (around 2-3%) in detection accuracy performance. These findings demonstrate the model's strong generalization capabilities and adaptability, thereby reinforcing its potential utility in varied IoT security scenarios. This comes from the fact that IGTD and self-attention models in the proposed model can build generalized patterns for transformed data from symmetric matrices of network traffic. However, we found that, while most DDoS activities can be identified, malicious traffic in highly specialized IoT environments with unique communication protocols (e.g., MQTT in DoS/DDoS-MQTT-IoT dataset) and rare attack patterns (e.g., Mirai botnet spreading codes target specific medical devices in Bot-IoT dataset), the model's performance suffer a degradation up to 5%. The degradation is eliminated after the model is trained with specialized datasets. This specific case highlights a potential limitation in generality if the dataset is small (i.e., CICDDoS2019). Generally, rich datasets (with well-prepared benchmarking distribution) are critical to maintaining generality advantage in vision transformer-based generative AI methods. Further, incremental learning [42] for real-time data collection can be another promising approach.

VI. Conclusion and Future Work

This paper presents a novel lightweight IDS that is particularly tailored for IoT devices. Initially, the system transforms tabular data into images while preserving spatial relationships. The method utilizes a vision transformer technique to reduce the network traffic feature dimensionality. The vision transformer encoder performs well with few learnable parameters and FLOPs. The evaluation results indicate that the proposed method can achieve outstanding accuracy scores of 99.62% and 99.55%, which are only marginally lower than the performance of existing methods by 0.07% and 0.04%. Meanwhile, the input dimensionality and the number of learnable parameters are reduced at 55.35% and 82.07%, respectively. Furthermore, there is a significant reduction in FLOPs by 93.56% and 99.28%.

In terms of model complexity, this study continuously demonstrates the most lightweight design, regardless of the extent to which the input dimensionality is decreased, compared to other current research. Regarding model size, when the input size stays the same, the proposed approach shows an expansion of 3.84 KB and 0.58 KB, which is slightly larger than other models. Nevertheless, this little increase results in substantial enhancements in accuracy, namely by 0.72% and 1.02%, respectively. In short, the evaluation results demonstrate the proposed approach's ability to balance model complexity, size, and performance. This capability makes the method ideal for resource-constrained IoT devices. Our goal for future work is to include the proposed method in an automated deep learning pipeline and few-shot learning to automatically create layers and increase detection performance in unseen IoT environments, even with sparse data.

References

[1] Google, "The largest L7 DDoS attack," in *https://cloud.google.com/blog/products/identity-security/how-google-cloud-blocked-largest-layer-7-ddos-attack-at-46-million-rps*, retrieved April 4, 2024.

[2] A. Jamalipour and S. Murali, "A Taxonomy of Machine-Learning-Based Intrusion Detection Systems for the Internet of Things: A Survey," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9444–9466, 2022.

[3] G. A. Mukhaini, M. Anbar, S. Manickam, T. A. Al-Amiedy, and A. A. Momani, "A systematic literature review of recent lightweight detection approaches leveraging machine and deep learning mechanisms in internet of things networks," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 1, p. 101866, 2024.

[4] C. Wang, D. Xu, Z. Li, and D. Niyato, "Effective Intrusion Detection in Highly Imbalanced IoT Networks With Lightweight S2CGAN-IDS," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[5] L. Yang, Y. Song, S. Gao, A. Hu, and B. Xiao, "Griffin: Real-time network intrusion detection system via ensemble of autoencoder in sdn," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2269–2281, 2022.

[6] H. Feng, W. Zhang, Y. Liu, C. Zhang, C. Ying, J. Jin, and Z. Jiao, "Multi-domain collaborative two-level ddos detection via hybrid deep learning," *Computer Networks*, vol. 242, p. 110251, 2024.

[7] H.-P. Tsai, V.-L. Nguyen, N. Chiewnawintawat, and R.-H. Hwang, "Sdt-ids: Spatial data transformation for elevating intrusion detection efficiency in iot networks," in *2024 IEEE International Conference on Communications Conference (ICC 2024), Denver, CO, USA*, 2024.

[8] M. Kravchik and A. Shabtai, "Efficient cyber attack detection in industrial control systems using lightweight neural networks and pca," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 4, pp. 2179–2197, 2022.

[9] S. I. Popoola, A. L. Imoize, M. Hammoudeh, B. Adebisi, O. Jogunola, and A. M. Aibinu, "Federated deep learning for intrusion detection in consumer-centric internet of things," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2023.

[10] R. Y. Aburasain, "Enhanced black widow optimization with hybrid deep learning enabled intrusion detection in internet of things-based smart farming," *IEEE Access*, vol. 12, pp. 16621–16631, 2024.

[11] F. A. A. Lins and M. Vieira, "Security requirements and solutions for iot gateways: A comprehensive study," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8667–8679, 2021.

[12] Y. Zhu, T. Brettin, F. Xia, A. Partin, M. Shukla, H. Yoo, Y. Evrard, J. Doroshow, and R. Stevens, "Converting tabular data into images for deep learning with convolutional neural networks," *Scientific Reports*, vol. 11, 05 2021.

[13] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*, pp. 1–8, 2019.
[14] S. M. Kasongo and Y. Sun, "A deep learning method with wrapper based feature extraction for wireless intrusion detection system," *Computers & Security*, vol. 92, p. 101752, 2020.
[15] G. Amaizu, C. Nwakanma, S. Bhardwaj, J. Lee, and D. Kim, "Composite and efficient ddos attack detection framework for b5g networks," *Computer Networks*, vol. 188, p. 107871, 2021.
[16] M. V. de Assis, L. F. Carvalho, J. J. Rodrigues, J. Lloret, and M. L. Proença Jr, "Near real-time security system applied to sdn environments in iot networks using convolutional neural network," *Computers & Electrical Engineering*, vol. 86, p. 106738, 2020.
[17] H. Asgharzadeh, A. Ghaffari, M. Masdari, and F. Soleimanian Gharehchopogh, "Anomaly-based intrusion detection system in the internet of things using a convolutional neural network and multi-objective enhanced capuchin search algorithm," *Journal of Parallel and Distributed Computing*, vol. 175, pp. 1–21, 2023.
[18] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for iot intrusion detection system," *Simulation Modelling Practice and Theory*, vol. 101, p. 102031, 2020. Modeling and Simulation of Fog Computing.
[19] N. Gupta, V. Jindal, and P. Bedi, "Lio-ids: Handling class imbalance using lstm and improved one-vs-one technique in intrusion detection system," *Computer Networks*, vol. 192, p. 108076, 2021.
[20] S. Zhu, X. Xu, J. Zhao, and F. Xiao, "LKD-STNN: A Lightweight Malicious Traffic Detection Method for Internet of Things Based on Knowledge Distillation," *IEEE Internet of Things Journal*, vol. 11, no. 4, pp. 6438–6453, 2024.
[21] M. He, Y. Huang, X. Wang, P. Wei, and X. Wang, "A Lightweight and Efficient IoT Intrusion Detection Method Based on Feature Grouping," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 2935–2949, 2024.
[22] W. Ding, M. Abdel-Basset, and R. Mohamed, "Deepak-iot: An effective deep learning model for cyberattack detection in iot networks," *Information Sciences*, vol. 634, pp. 157–171, 2023.
[23] O. R. Sanchez, M. Repetto, A. Carrega, R. Bolla, and J. F. Pajo, "Feature selection evaluation towards a lightweight deep learning DDoS detector," in *IEEE International Conference on Communications*, 2021.
[24] Y. Wei, J. Jang-Jaccard, F. Sabrina, A. Singh, W. Xu, and S. Camtepe, "AE-MLP: A Hybrid Deep Learning Approach for DDoS Detection and Classification," *IEEE Access*, vol. 9, pp. 146810–146821, 2021.
[25] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martínez-del Rincón, and D. Siracusa, "Lucid: A practical, lightweight deep learning solution for DDoS attack detection," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 876–889, 2020.
[26] R. Zhao, G. Gui, Z. Xue, J. Yin, T. Ohtsuki, B. Adebisi, and H. Gacanin, "A novel intrusion detection method based on lightweight neural network for internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9960–9972, 2022.
[27] A. Zainudin, L. A. C. Ahakonye, R. Akter, D.-S. Kim, and J.-M. Lee, "An efficient hybrid-dnn for DDoS detection and classification in software-defined iiot networks," *IEEE Internet of Things Journal*, vol. 10, no. 10, pp. 8491–8504, 2023.
[28] C. Yao, Y. Yang, K. Yin, and J. Yang, "Traffic anomaly detection in wireless sensor networks based on principal component analysis and deep convolution neural network," *IEEE Access*, vol. 10, 2022.
[29] M. M. Alani and A. I. Awad, "An intelligent two-layer intrusion detection system for the internet of things," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 683–692, 2023.
[30] S. I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui, and H. Gacanin, "Hybrid deep learning for botnet attack detection in the internet-of-things networks," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4944–4956, 2021.
[31] M. Abdel-Basset, H. Hawash, K. M. Sallam, I. Elgendi, K. Munasinghe, and A. Jamalipour, "Efficient and lightweight convolutional networks for iot malware detection: A federated learning approach," *IEEE Internet of Things Journal*, vol. 10, no. 8, pp. 7164–7173, 2023.
[32] K. He, W. Zhang, X. Zong, and L. Lian, "Network intrusion detection based on feature image and deformable vision transformer classification," *IEEE Access*, vol. 12, pp. 44335–44350, 2024.
[33] B. Wang, Y. Su, M. Zhang, and J. Nie, "A deep hierarchical network for packet-level malicious traffic detection," *IEEE Access*, vol. 8, pp. 201728–201740, 2020.
[34] L. Sana, M. M. Nazir, J. Yang, L. Hussain, Y.-L. Chen, C. S. Ku, M. Alatiyyah, S. A. Alateyah, and L. Y. Por, "Securing the iot cyber environment: Enhancing intrusion anomaly detection with vision transformers," *IEEE Access*, vol. 12, pp. 82443–82468, 2024.
[35] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.
[36] A. Alatram, L. F. Sikos, M. Johnstone, P. Szewczyk, and J. J. Kang, "DoS/DDoS-MQTT-IoT: A dataset for evaluating intrusions in IoT networks using the MQTT protocol," *Computer Networks*, vol. 231, p. 109809, 2023.
[37] A. Habibi Lashkari., G. Draper Gil., M. S. I. Mamun., and A. A. Ghorbani., "Characterization of tor traffic using time based features," in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy - ICISSP*, pp. 253–262, SciTePress, 2017.
[38] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.
[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.
[40] L. Beyer, X. Zhai, and A. Kolesnikov, "Better plain vit baselines for imagenet-1k," 2022.
[41] S. Yuan, H. Li, R. Zhang, M. Hao, Y. Li, and R. Lu, "Towards lightweight and efficient distributed intrusion detection framework," in *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2021.
[42] L. Du, Z. Gu, Y. Wang, L. Wang, and Y. Jia, "A few-shot class-incremental learning method for network intrusion detection," *IEEE Transactions on Network and Service Management*, vol. 21, no. 2, pp. 2389–2401, 2024.

Van-Linh Nguyen (Member, IEEE) is an Assistant Professor at the Department of Computer Science and Information Engineering, National Chung Cheng University (CCU), Taiwan, and the lead of the Cyber Information Security Laboratory (CIS Lab). His research interests include cybersecurity, network intelligence, wireless communications, and vehicular networks.

Hao-Ping Tsai received a BS and MSc degree at the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan, in 2022 and 2024, respectively. His research interests include cybersecurity, automated deep learning, and Internet of Things networks.

Hyundong Shin (Fellow, IEEE) is a Professor at Kyung Hee University, Korea. His research interests include quantum information science, wireless communication, and machine intelligence. He received the IEEE Communications Society's Guglielmo Marconi Prize Paper Award and William R. Bennett Prize Paper Award. He served as a Publicity Co-Chair for IEEE PIMRC and a Technical Program Co-Chair for IEEE WCNC and IEEE GLOBECOM. He was an Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and IEEE COMMUNICATIONS LETTERS.

Trung Q. Duong (Fellow, IEEE) is a Canada Excellence Research Chair (CERC) and a Full Professor at Memorial University of Newfoundland, Canada. He is also the adjunct Chair Professor in Telecommunications at Queen's University Belfast, UK and a Research Chair of Royal Academy of Engineering. His current research interests include quantum communications and wireless communications.