IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received October 17, 2021, accepted October 25, 2021, date of publication October 27, 2021, date of current version November 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3123791

AE-MLP: A Hybrid Deep Learning Approach for DDoS Detection and Classification

YUANYUAN WEI [1], JULIAN JANG-JACCARD [1], FARIZA SABRINA [2], (Member, IEEE), AMARDEEP SINGH [1], WEN XU [1], AND SEYIT CAMTEPE [3], (Senior Member, IEEE)

[1]Cybersecurity Laboratory, Comp Sci/Info Tech, Massey University, Auckland 0632, New Zealand
[2]School of Engineering and Technology, Central Queensland University, Sydney, NSW 2000, Australia
[3]Data61, CSIRO, Marsfield, NSW 2015, Australia

Corresponding author: Yuanyuan Wei (y.wei1@massey.ac.nz)

This work was supported by the Cyber Security Research Programme—Artificial Intelligence for Automating Response to Threats—from the Ministry of Business, Innovation, and Employment (MBIE) of New Zealand as a part of the Catalyst Strategy Funds under Grant MAUX1912.

ABSTRACT Distributed Denial-of-Service (DDoS) attacks are increasing as the demand for Internet connectivity massively grows in recent years. Conventional shallow machine learning-based techniques for DDoS attack classification tend to be ineffective when the volume and features of network traffic, potentially carry malicious DDoS payloads, increase exponentially as they cannot extract high importance features automatically. To address this concern, we propose a hybrid approach named AE-MLP that combines two deep learning-based models for effective DDoS attack detection and classification. The Autoencoder (AE) part of our proposed model provides an effective feature extraction that finds the most relevant feature sets automatically without human intervention (e.g., knowledge of cybersecurity professionals). The Multi-layer Perceptron Network (MLP) part of our proposed model uses the compressed and reduced feature sets produced by the AE as inputs and classifies the attacks into different DDoS attack types to overcome the performance overhead and bias associated with processing large feature sets with noise (i.e., unnecessary feature values). Our experimental results, obtained through comprehensive and extensive experiments on different aspects of performance on the CICDDoS2019 dataset, demonstrate both a very high and robust accuracy rate and F1-score that exceed 98% which also outperformed the performance of many similar methods. This shows that our proposed model can be used as an effective DDoS defense tool against the growing number of DDoS attacks.

INDEX TERMS Distributed denial of service, DDoS, deep learning, multi-class classification, autoencoder, MLP, CICDDoS2019.

I. INTRODUCTION

A Distributed Denial-of-Service (DDoS) attack occurs when attackers make victims' system/network resources unavailable by sending massive amounts of requests to flood the system resources/bandwidth of the victim's system [1], [2]. Typically, attackers exploit vulnerabilities in transport, network, and application layer protocols (e.g., TCP, UDP, HTTP, and ICMP, etc.) [3], [4] to send malicious payloads (e.g., network packets). With the demand for Internet connectivity expand rapidly to mobile devices and Internet of Things (IoT) which is predicted to reach 500 billion by 2030 [5], there is an increasing concern for developing effective DDoS defense techniques. To address this concern, many Artificial Intelligence (AI) methods, both using traditional shallow machine learning-based and more advanced deep neural network-based, have been proposed to demonstrate the feasibility of such AI-based approaches to safeguard our networks from DDoS attacks. One of the essential tasks in proposing the next generation of DDoS defense techniques is with the effectiveness of feature extraction techniques. Because it is infeasible and expensive to analyze the entire raw network traffic samples manually among the large feature sets when not all of them provide useful information for detecting malicious payloads. [22], [23]. Many state-of-the-art have proposed solutions to feature extraction for DDoS attack detection and classifications using different feature extraction methods [2], [4], [22]. Though these existing

The associate editor coordinating the review of this manuscript and approving it for publication was Diego Oliva.

TABLE 1. The summary of existing ML and DL-based approaches.

Paper	Techniques	Domain	Performance	Dataset
Maseer et al. [6]	KNN, NB, RF, SVM	IDS	98.86 ≈ 99.54 (Accuracy)	CICDDoS2017
Ullah et al. [7]	NB, LR, DT, RF	IoT	99.99 ≈ 100 (F1-score)	IoT Botnet
Gohil et al. [8]	DT, NB, LR, SVM, KNN	IDS	97.72 ≈ 99.99 (Accuracy)	CICDDoS2019
Alamri et al. [9]	XGBoost	SDN	99.9 (Accuracy)	CICDDoS2019
Khoei et al. [10]	Stacking, Bagging, Boosting	Smart Grid	92.2 ≈ 93.4 (Accuracy)	CICDDoS2019
Parfenov et al. [11]	Gradient Boosting	IDS	96.8 (F1-score)	CICDDoS2019
Parfenov et al. [11]	CatBoost	IDS	96.9 (F1-score)	CICDDoS2019
Sanchez et al. [12]	Random Forest	IoT	99.97 (Accuracy)	CICDDoS2019
Varghese et al. [13]	D3	SDN	84.54 (Accuracy)	CICDDoS2019
Pontes et al. [14]	EFC	IDS	97.5 (F1-score)	CICDDoS2019
Shieh et al. [15]	Bi-directional LSTM + Gaussian Mixture	IDS	98 (Accuracy)	CICDDoS2019
Sanchez et al. [12]	MLP	IDS	99.93 (Accuracy)	CICDDoS2019
Rehman et al. [16]	Gated Recurrent Units (GRU)	IDS	99.69 ≈ 99.94 (Accuracy)	CICDDoS2019
Almaini et al. [17]	Kalman Backpropagation Neural Network	IDS	94 (Accuracy)	CICDDoS2019
Samom and Taggu [18]	MLP	IDS	99.92 (Accuracy)	CICDDoS2019
Elsayed et al. [19]	RNN + Autoencoder	SDN	99 (F1-score)	CICDDoS2019
Javaid et al. [20]	Sparse-Autoencoder	IDS	88.39 (Accuracy)	NSL-KDD
Sadaf et a. [21]	Autoencoder	IDS	88.98 (Accuracy)	NSL-KDD
Can et al. [2]	MLP	IDS	79.39 (F1-score)	CICDDoS2019

shallow machine learning (ML) approaches have been shown to achieve high detection accuracy, some limitations have been discussed. For instance, Nguyen and Reddi [24] pointed out the inefficiency in using ML approaches in handling raw, unlabelled, or high dimensional data. Others [25]–[27] indicated that the accuracy of detection degrades with ML approaches when a large dataset requires some level of manual feature extraction. Effective feature extraction of network traffic samples that are most relevant to the detection and classification task does not only increases high accuracy but also can accelerate the execution time to analyze the data. In this study, we propose a hybrid deep learning technique that utilizes two deep neural network models for effective feature extraction and accurate DDoS attack detection and classification without human intervention. The contribution of our proposed model is summarized as follows:

- We propose a hybrid deep learning model named "AE-MLP" not only to detect DDoS attacks but also classify the attack into different DDoS attack types in a timely manner.
- Our proposed model uses an Autoencoder (AE) to extract the most important features from a large-scale DDoS attack dataset. Finding the set of compressed and reduced feature sets, most relevant to detect malicious payload, not only improves the accuracy of detection but can also effectively reduce expensive execution time.
- Our proposed model does not only detect potential DDoS attacks but can effectively classify different DDoS attack types. This classification capability can provide an opportunity for cybersecurity professionals to devise an optimal and relevant response strategy as quickly as possible before disastrous damage is done by different DDoS attack types.
- Our experimental results, comprehensively and extensively evaluated, demonstrate a very high and robust F1-score over 98% for detecting DDoS attacks and classifying them into correct attack types. Our results outperformed the performance of many similar methods.

We organize the rest of the paper as follows. Section II examines the related work. Section III provides the details of the proposed AE-MLP model that contains the feature extraction and classification strategies as well as the algorithm involved. Section IV illustrates the details of the dataset we used in our study and the methodologies we used for data pre-processing. We describe the experimental results in Section V including the experimental setup, the performance metrics we used, performance of our proposed model, and a comparison to other similar models. Section VI provides a conclusion of our work and future work directions.

II. RELATED WORK

We review the existing state-of-the-art in addressing DDoS detection and classification using Artificial Intelligence techniques, both shallow machine learning and deep learning-based neural network here. The summary of these related works are shown in Table 1.

A. MACHINE LEARNING-BASED APPROACHES

Many classical shallow machine learning techniques have been used for DDoS classification. The authors in [6]–[8] presents the performance of many classic machine learning techniques, such as Naïve Bayes, Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, and K-Nearest-Neighbour against various DDoS datasets with the detection accuracy reaching near 99%.

Ensemble-based techniques were proposed by [9]–[11] by utilizing different techniques involved in bagging, boosting, and stacking, and the results show that some of these techniques outperforming Random forest, Naïve Bayes, and KNN in detecting DDoS attacks in different application contexts (e.g., Smart Grid, IoT).

The authors [12] propose a lightweight approach to detect DDoS attacks aimed at resource-constrained environments such as IoT and shows that their lightweight random forest technique can achieve as high as 99% of detection accuracy. Varghese and Muniyal [13] proposed a statistical anomaly detection algorithm implemented in the data plane of Software Defined Network (SDN) to detect DDoS attacks near real-time as a part of an Intrusion Detection System (IDS). Pontes *et al.* [14] propose an Energy-based Flow Classifier (EFC) which utilizes inverse statistics to infer anomaly scores base on labeled benign examples. The anomaly scores are then used as classifying different DDoS attacks. Their approach achieved 97.5% F1-score while the outcomes of other performance metrics were not presented.

Despite often very high detection rate that achieves 99% accuracy, however, many argued [24]–[27] that the detection accuracy degrades with the increase of the size of dataset often containing high dimensional features. In addition, these approaches become impractical when they require raw or unlabelled datasets that require manual feature extraction. To address this limitation, a number of deep learning-based neural networks that can detect DDoS attacks have been proposed.

B. DEEP LEARNING-BASED APPROACHES

Shieh *et al.* [15] demonstrate a Bi-directional LSTM model along with a Gaussian Mixture Model to detect and classify 6 different types of DDoS attacks with an accuracy of 98%. Sanchez *et al.* [12] proposed a standalone Multi-Layer Perceptron (MLP) achieving the 99.93% accuracy and 99.96% F1-score. Rehman *et al.* [16] proposed a Gated Recurrent Units (GRU) model to detect DDoS attack based on CICDDoS2019 dataset. They achieved the highest accuracy of 99.69% for reflection attacks and 99.94% for exploitation attacks. Almaini *et al.* [17] proposed Kalman Backpropagation Neural Network where the Kalman algorithm is used to fine-tune weight metrics while backpropagation was utilized to tune biases. The performance evaluation results of their proposed model achieved the performance of 94% accuracy with a low false alarm rate (0.0952). Samom and Taggu [18] proposed an MLP model to detect 4 different DDoS attack types (i.e., SYN, NET, Portmap, and UDPLag) and compares the results with other machine learning methods. Their study uses the Chi-Squared Function as a feature extractor to select 20 features then uses the PCA technique for dimension reduction. Their proposal showed that their model achieved 99.92% accuracy on the CICDDoS2019 dataset. Though some of these existing works appear to provide good performance near 99%, they often only offer binary classification where it only detects whether network traffic contains a DDoS attack or not but don't offer to classify what type of DDoS attack it is.

Elsayed *et al.* [19] proposed a hybrid method named DDoSNet that combines a Recurrent Neural Network (RNN) with an Autoencoder to detect DDoS attack at the Software-Defined Networking (SDN) layer. The evaluation result of their proposal showed that the DDoSNet model achieved the highest performance metrics based on Confusion Matrix but again they also offer only binary classification. Javaid *et al.* [20] proposed sparse-autoencoder for feature learning and soft-max regression-based neural architecture for classification and they achieved 88.39% accuracy. The authors in [21] automated threshold learning for anomaly detection in an autoencoder-based model by combining it with unsupervised learning technique isolation forest and got 88.98% accuracy. Can *et al.* [2] proposed DDoSNet which utilizes an automatic Feature Selection (FS) technique based on the context of the whole feature set then classifies them with fully connected MLP. This proposed method achieved 91.16% precision, 79.41% recall, and 79.39% F1-score for multi-class classification. The authors emphasized that the limitation of their approach was with low performance as they were using the whole feature set for classification.

III. OUR MODEL

Our proposed AE-MLP model consists of two phases: 1) The first phase involves feature extraction via an AE; 2) The second phase involves DDoS attack type classification via an MLP. During the first phase, we build an AE model by using traffic samples as input to train the model. Once AE is trained, the features from the bottleneck layer are extracted. These extracted features from the bottleneck layer of the AE model are fed as inputs to the second phase where MLP uses it to classify different DDoS attack types. Figure 1 illustrates the overall approach that is used by our proposed AE-MLP algorithm.

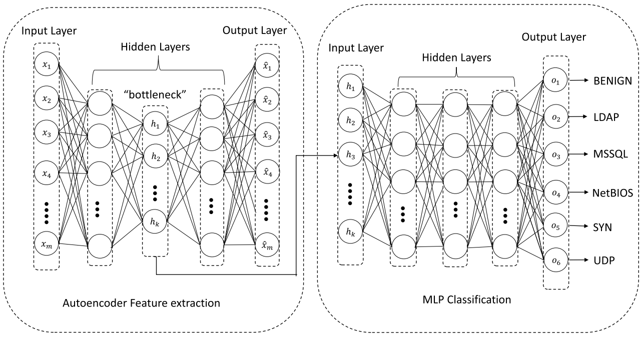


FIGURE 1. The overview of AE-MLP.

A. AUTOENCODER AND FEATURE EXTRACTION

Our AE is an unsupervised feed-forward neural network. It composes of an input layer, an output layer, and several hidden layers. It has a symmetrical pattern – the output layer has the same number of neurons as the input layer while any hidden layer generally has fewer neurons than the input and output layer. The bottleneck layer, also referred to as a latent space, is one of the hidden layers which has the smallest number of neurons. The latent space contains the compressed representation of the input. Typically, the main aim of AE is to reconstruct the input from the output, i.e. $\hat{x} \approx x$ where x indicates the input while $\hat{x}$ indicates the output. In our approach, the input is reconstructed using AE while doing so we get the hidden layer feature embedding with the minimum number of neurons which represents a lower-dimensional projection of input features. This hidden layer embedding captures data characteristics in the lower dimension, thus we have used it as input features to the MLP classification model. To use AE as a feature extraction engine, our model goes through the following steps.

1) ENCODING

In the encoding operation (E_ϕ), any input sample x is a m dimensional vector ($x \in \mathbb{R}^m$) and is mapped to the bottleneck layer representation (h), as shown in Equation (1).

$$h = f_1(w_1 x + b_1) \tag{1}$$

where w_1 is the weight matrix, b_1 is a bias and f_1 is an activation function.

2) DECODING

In the decoding operation (D_θ), the bottleneck layer representation of (h) is mapped back into a reconstruction of x, as shown in Equation (2):

$$\hat{x} = f_2(w_2 h + b_2) \tag{2}$$

where f_2 is an activation function for the decoder. w_2 is the weight matrix, b_2 represents a bias and $\hat{x}$ represent reconstructed input sample.

3) LOSS FUNCTION

To minimize reconstruct error on x with non-linear functions, the loss reconstruction (L) is calculated from Equation (3).

$$L(x, \hat{x}) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2 \tag{3}$$

where n represents the number of training samples.

4) FEATURE EXTRACTION

The equations (1), (2) and ((3)) represent working of single hidden layer auto-encoder where h represent bottleneck layer feature embedding (encoding) of AE. This embedding size is dependent on number of neurons (k) in hidden layer, in general its size is less compare to input dimension ($k \ll m$). AE model applies backpropagation to obtain optimal values for the weight matrix $w_1 \in R^{m \times k}$ and $w_2 \in R^{k \times m}$ and bias $b_1 \in R^{m \times 1}$ and $b_2 \in R^{k \times 1}$ in equations (1) and (2) respectively to minimize the difference between input x and output $\hat{x}$. Mostly rectified linear unit (ReLU) is used as non-linear activation function in the hidden layer. In practice, multiple hidden layers are used, where each layer have its own encoding and decoding function described in equation (1) and (2) respectively. In our work, we have used hidden layer with lowest number of neurons as feature vector for our Multi-layer Perceptron classification model.

B. MULTI-LAYER PERCEPTRON NETWORK AND CLASSIFICATION

MLP is also a feed-forward network, unlike AE its output layer is equal to number of classes (p). The MLP has input layer (our case equal to size of AE bottleneck layer), multiple hidden layers and output layer (equal to sum of attacks and benign classes, $p = 6$). Similar to AE, hidden layer use non-linear activation function (in our case, we used ReLU function) to extract information from input features as shown in Equation (4)

$$y_z = f_{relu}(h_z w_j + b_j) \tag{4}$$

where h_z represent latent space embedding from AE as feature vector, w_j is the weight matrix, b_j is bias vector, f_{relu} non-linear activation function of hidden layer and y_z is information extracted at hidden layer of MLP.

By processing the input vector h_z, the hidden layers of MLP produce the vector y_z. This vector y_z is fed as input to output layer to predict output class. The output layer mostly use softmax function for multiclass problem. Finally, output class ($\hat{y} \in \mathbb{R}^p$) can be predicted using equation (5).

$$\hat{y} = softmax(y_z w_y + b_y) \tag{5}$$

where w_y and b_y are weights matrix and bias vector for the output layer.

C. AE-MLP ALGORITHM

The algorithm for our proposed AE-MLP model for DDoS attack detection and classification is shown in Algorithm 1.

In our proposed model, we use AE as a feature extraction tool that can transform the original data (e.g., network traffic) from the high dimensional space to the non-linear low dimensional space. By doing this, the latent space at the AE now contains the number of features that can be best represented to detect if network traffic contains a malicious DDoS payload and further classify what type of DDoS attack payload it carries. To determine the best features for DDoS detection, our AE goes through the following steps:

- We first use the unsupervised learning mode of the AE to train on the training dataset for dimensionality reduction purposes.
- We have experimented on the number of different AE architecture in terms of the number of input, hidden, and output layers and corresponding hyperparameters. The best optimized AE architecture was the one that uses 77 encoded features as input, a single hidden layer with 32 neurons, and the latent space that represents the 24 features as the last hidden layer.
- The 24 features at the latent space are extracted.

The extracted features are then fed into the MLP model as inputs and are now used to train the MLP model as a classifier to detect different DDoS attack types. To classify different DDoS attack types, our MLP goes through the following steps:

Algorithm 1 AE-MLP Classification

```
Input: Training dataset $X = \{x_1, x_2, x_3, \ldots, x_n\}$
Testing dataset $X' = \{x'_1, x'_2, x'_3, \ldots, x'_n\}$
Training Label $Y = \{y_1, y_2, \ldots, y_n\}$
Testing Label $Y' = \{y'_1, y'_2, \ldots, y'_n\}$
Encoder $E_\phi$; Decoder $D_\theta$; MLP $M_\delta$
Output: $O(\hat{y'}|y')$
begin
    /* Phase 1: AE feature extraction */
    /* Training AE in mini-batch */
    $\phi, \theta \leftarrow$ Initialize parameters
    for number of training iterations do
        sample mini-batch of $k$ samples
        $\{X_1, X_2, X_3, \ldots, X_k\}$ from $X$
        /* Calculating loss */
        $V(E, D) = \frac{1}{k}\sum_{i=1}^{k}(X_i - D_\theta(E_\phi(X_i)))^2$
        $\phi, \theta \leftarrow$ Update parameters using Stochastic
        Gradient Descent of V
    end
    /* Phase 2: MLP Classification */
    $\delta \leftarrow$ Initialize parameters
    /* Training MLP */
    for each $(x, y) \in (X, Y)$ do
        /* get latent presentations from
        trained AE */
        $a \leftarrow E_\phi(x)$
        /* trained MLP $M_\delta$ */
        $O(\hat{y}|y) \leftarrow M_\delta((a), y)$
        $\delta \leftarrow$ Update parameters using Stochastic
        Gradient Descent
    end
    /* Testing */
    for each $(x', y') \in (X', Y')$ do
        $a' \leftarrow E_\phi(x')$
        $O(\hat{y'}|y') \leftarrow M_\delta((a'), y')$
    end
end
```

- We use the supervised learning mode of the MLP to train on the training dataset using the label contained in the training dataset.
- We have experimented on the number of different MLP architectures. The best optimized MLP architecture was the one that uses 5-layers – 1 input layer, 3 hidden layers, and 1 output layer.
- The activation function "relu" was used for hidden layers while "softmax" function was used at the last output layer for classification.

IV. DATA AND METHODOLOGIES

In this section, we provide the details of the data we used for our study, the methodology we employed for data processing, and the workflow of our proposed model. The CIC-DDOS dataset has two datasets, training and testing datasets, respectively. As seen in Figure 2, we first use only the training dataset after applying a number of data pre-processing techniques such as data cleaning, removing irrelevant features, label encoding, and normalizes the dataset by scaling them to fit in the range of [0, 1]. After pre-processing the training dataset, we fit the dataset into our proposed model for AE training and subsequent feature extraction. The extracted features are then fed into the MLP. Another training by MLP proceeds to train the MLP model. Once our proposed model is well trained, we use the testing dataset first fed into the AE for the feature extraction using the hyperparameters that were trained during the AE training phase, the extract features then are fed into the trained MLP for a classification task to categorize different DDoS attack types.

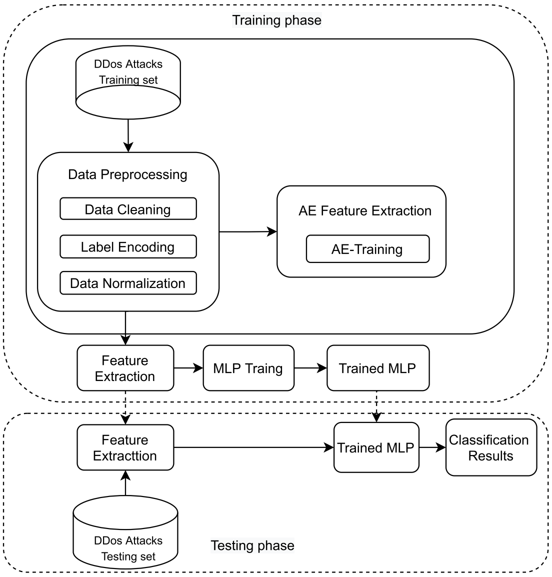


FIGURE 2. AE-MLP classification.

A. CICDDoS2019 DATASET

In this study, we use CICDDoS2019 [4] dataset that has been widely used for DDoS attack detection and classification. The dataset contains a large amount of up-to-date realistic DDoS attack samples as well as benign samples. The total number of records contained in CICDDoS2019 is depicted in Table 2.

TABLE 2. The number of records in CICDDoS2019.

dataset	total	benign	malicious
Training day	50,063,112	56,863	50,006,249
Testing day	20,364,525	56,965	20,307,560

Each record of the dataset contains 88 statistical features (e.g., timestamp, source and destination IP addresses, source and destination port numbers, the protocol used for the attack, and a label for a type of DDoS attack). The training dataset contains a total of 12 different DDoS attacks (i.e., NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN, and TFTP) while only 7 DDoS attacks are included in the testing dataset (i.e., PortScan, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag and SYN). The details of

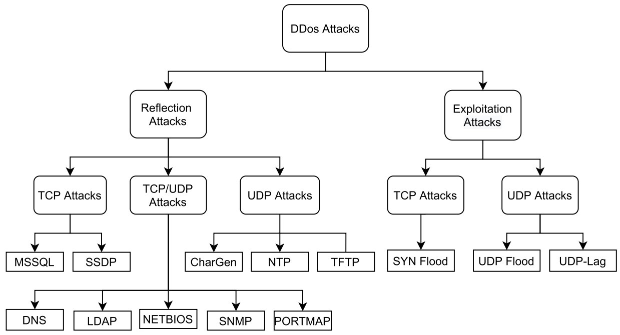


FIGURE 3. DDos attacks categorization and hierarchy [4].

the date and number of records collected for different DDoS attack types are present in Table 3.

TABLE 3. Daily label of data collection.

Days	Attacks	Attack times	Flow Count
Training Set	LDAP	11:22 - 11:32	2,179,930
	MSSQL	11:36 - 11:45	4,522,492
	NetBIOS	11:50 - 12:00	4,093,279
	UDP	12:45 - 13:09	3,134,645
	SYN	13:29 - 13:34	1,582,289
Testing Set	NetBIOS	10:00 - 10:09	3,657,497
	LDAP	10:21 - 10:30	1,915,122
	MSSQL	10:33 - 10:42	5,787,453
	UDP	10:53 - 11:03	3,867,155
	SYN	11:28 - 17:35	4,891,500

These DDoS attacks cover two different categories, some belong to Reflection-based and others belong to Exploitation-based.

1) REFLECTION-BASED ATTACKS

The attackers of the DDoS attack in this category typically send malicious network packets sent to reflector servers with the source IP address set to target the victim's IP address so that the victim is overwhelmed to send an enormous number of response packets. These attacks are typically carried out through application layer protocols. In terms of our CICD-DoS2019 dataset, any traffic with the (application layer) protocol defined for MSSQL, SSDP, NTP, TFTP, DNS, LDAP, NetBIOS, and SNMP be reflection-based attacks.

2) EXPLOITATION-BASED ATTACKS

The attackers of the DDoS attack in this category exploit a particular protocol used in the network, transport, and application level of the Open Systems Interconnection (OSI) model or TCP/IP 5-layer model. Transport layer protocols such as TCP or UDP are typically used to overwhelm the victim's IT resources (e.g., SYN flood, UDP flood, and UDP-Lag) by sending a massive number of TCP or UDP packets. The dataset labeled with SYN, UDP, and UDP-lag in CIDDDOS2019 belongs to this category. The DDoS attack categorization is seen in Table 3

In our study, we use 5 DDoS attack types (i.e., LDAP, MSSQL, NetBIOS, SYN, UDP) and benign traffic samples to train and test our proposed model. The high-level description of the nature of the DDoS attack used in our study is summarised as follows.

- *LDAP Attack:* In this DDoS attack, an application layer protocol, Lightweight Directory Access Protocol LDAP) typically used to obtain a human-readable URL (e.g., google.com), is exploited by an attacker to send requests to a publicly available but vulnerable LDAP server to generate large responses.
- *NetBIOS Attack:* In this DDoS attack, Network Basic Input/Output System (NetBIOS) is exploited by an attacker which sends spoofed ''Name Release'' or ''Name Conflict'' messages to a victim machine in order to refuse all NetBIOS network traffic.
- *MSSQL Attack:* An attacker exploits the vulnerabilities in Microsoft Structured Query Language (MSSQL) where the attacker pretended to be a legitimate MSSQL client by executing the scripted requests using a forged IP address to the MSSQL Server to appear as coming from the target server.
- *SYN Attack:* The SYN flood attack exploits the TCP-three-way handshake by sending a massive number of repeated SYN packets to the target machine until the server crashes/malfunctions.
- *UDP Attack:* In the UDP flood attack, UDP packets are sent to random ports on the target machine at a very high rate. As a result, the available bandwidth of the network gets exhausted, system crashes and performance degrades. The firewall protecting the target server can be exhausted as a result.

B. DATA PRE-PROCESSING

In this section, we discuss the methodologies we used to process our dataset to feed into our proposed AE-MLP model.

1) DATA CLEANING

The original dataset contained 88 features. As suggested by [17], we also removed the features not contributing to detect DDoS attacks. These include the feature such as ''Unnamed'', ''Flow ID'', ''Source IP'', ''Destination IP'', ''Source Port'', ''Destination Port'', ''Timestamp'', ''Flow Bytes'', ''Flow Packets'', and ''SimilarHTTP''. After the exclusion of these 10 features, we have 78 features to work with. Following the recommendation of the work by [18], we further cleaned up the values containing NaN (not a number), blank, and infinity values to set 0.

2) LABEL ENCODING

We had to substitute the categorical labels as deep models only operate on float/numeric values. One categorical value we had to convert was the attack label (i.e., benign and the five attack types). We used a 6-bit feature vector to indicate different labels, for example $[1, 0, 0, 0, 0, 0]$ indicates benign, $[0, 1, 0, 0, 0, 0]$ indicates LDAP attack type, and [0, 0, 1, 0, 0, 0] indicates MSSQL attack, etc. With an additional 6 feature vectors added, we had a total of 83 features (i.e., 77 representing the original features plus 6 features for an attack label).

3) DATA NORMALIZATION

The CICDDoS2019 datasets contain some features with very high variance in terms of value between the minimum and the maximum (e.g., ''Flow Duration'', ''Flow IAT Std'', ''Flow IAT Max'', ''Bwd IAT min''). We applied a normalization strategy to eliminates the impacts of big variance of the values across the features thus reduces the execution time for model training and improving accuracy. There are several widely used methods to perform feature scaling, including Z Score, standardization, normalization. As proposed by [16], we use MinMax-based normalization for our feature scaling. This method maps the original range of each feature into a new range with Equation (6)

$$Z_i = \frac{Z_i - min}{max - min} \tag{6}$$

where Z_i donates all the normalized numeric values ranging between [0-1]; *max* and *min* donates the maximum and minimum values from all data points.

V. EXPERIMENTAL RESULTS

In this section, we provide the details of the experiment including the environment setup, analysis of results, and discussion.

A. EXPERIMENT SETUP

Our experiments were carried out using the following system setup shown in Table 4.

TABLE 4. Implementation environment specification.

Unit	Description
Processor	3.4GH$_z$ Inter Core i5
RAM	16GB
OS	MacOS Big Sur 11.4
Packages used	tensorflow 2.0.0, sklearn 0.24.1

To evaluate the performance of our proposed model, we use the classification accuracy, precision, recall, and F1 score as performance metrics. Table 5 illustrates the confusion matrix, where:

- True Positive (TP) indicates anomalous traffic correctly classified as anomalous.
- True Negative (TN) indicates normal traffic correctly classified as normal.
- False Positive (FP) indicates normal traffic incorrectly classified as anomalous.
- False Negative (FN) indicates anomalous traffic incorrectly classified as normal.

Based on the aforementioned terms, the evaluation metrics are calculated as follows.

TABLE 5. Confusion matrix.

Total Population		Predicted Condition	
		Normal	Anomaly
Actual Condition	Normal	TN	FP
	Anomaly	FN	TP

True Positive Rate (also known as Recall) estimates the ratio of the correctly predicted samples of the class to the overall number of instances of the same class. It can be computed using Equation (7). Higher TPR $\in$ [0, 1] value indicates the good performance of the machine learning model.

$$TPR(Recall) = \frac{TP}{TP + FN} \tag{7}$$

False Positive Rate (FPR) presents the proportion of data points correctly classified as anomalous, which can be calculated in Equation (8).

$$FPR = \frac{FP}{FP + TN} \tag{8}$$

Precision (Pre) measures the quality of the correct predictions. Mathematically, it is the ratio of correctly predicted samples to the number of all the predicted samples for that particular class as shown in Equation (9). Precision is usually paired with Recall to evaluate the performance of the model. Sometimes pair can appear contradictory thus comprehensive measure F1-score is considered.

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

F1-Score computes the trade-off between precision and recall. Mathematically, it is the harmonic mean of precision and recall as shown in Equation (10).

$$F1 = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall} \right) \tag{10}$$

Accuracy (Acc) measures the total number of data samples correctly classified, as shown in Equation (11).

$$A_{CC} = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

The area under the curve (AUC) computes the area under the receiver operating characteristics (ROC) curve which is plotted based on the trade-off between the true positive rate on the y-axis and the false positive rate on the x-axis across different thresholds. Mathematically, AUC is computed as shown in Equation (12).

$$AUC_{ROC} = \int_0^1 \frac{TP}{TP + FN} d \frac{FP}{TN + FP} \tag{12}$$

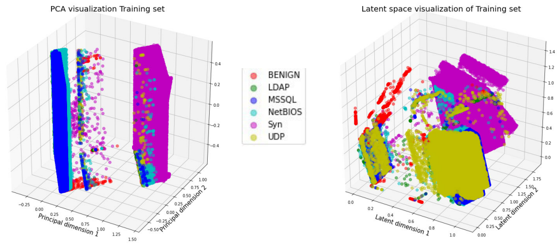
The training parameters are shown in Table 6.

B. PERFORMANCE OF OUR PROPOSED MODEL

We used 5% of the original CICDDoS2019 dataset from the day one collection for training as it was not feasible to use the full dataset due to performance consideration. Figure 4 shows the PCA and latent space visualizations of the

TABLE 6. Training parameters.

Algorithms	Hyperparameters	values
AE	Mini-batch Learning rate N-iterations Epoch	32 0.001 19137 20
MLP	activation solver hidden layers size	relu adam [23,15,10]



(a) Training set visualisation

(b) Training latent visualisation

FIGURE 4. The PCA visualization of training set and latent space visualization (bottleneck layer of AE) of training set.

training dataset. In the PCA visualization, it is very difficult to identify different clusters of attacks, while the latent space components can identify most clusters. This exploratory analysis suggests that the latent space embedding better captures the features in a lower dimension. We can think of latent space as a coordinate system in which similar points are placed together. Therefore, feature embedding from the hidden layer of AE is more suitable for classification than raw features, as can be seen from the classification results. We would also like to point out that although the PCA components do not show a complete picture of the feature space, they still increase the interpretability of the data. From Figure 4, it is visible that there is no balance in the number of traffic samples between benign (red color) and DDoS attack (other colors) as the number of benign is significantly less.

Our model used 80% of this data as the training dataset to fine-tune the model while 20% was used as a validation dataset to fine-tune the model's hyperparameters. We used 5% of the original CICDDoS2019 dataset for each subset from the day two collection as the testing dataset.

In our study, we limit the classification of benign and 5 DDoS attacks - LDAP, MSSQL, NetBIOS, SYN, and UDP – during the testing phase to avoid any implication of biases due to an imbalanced dataset.

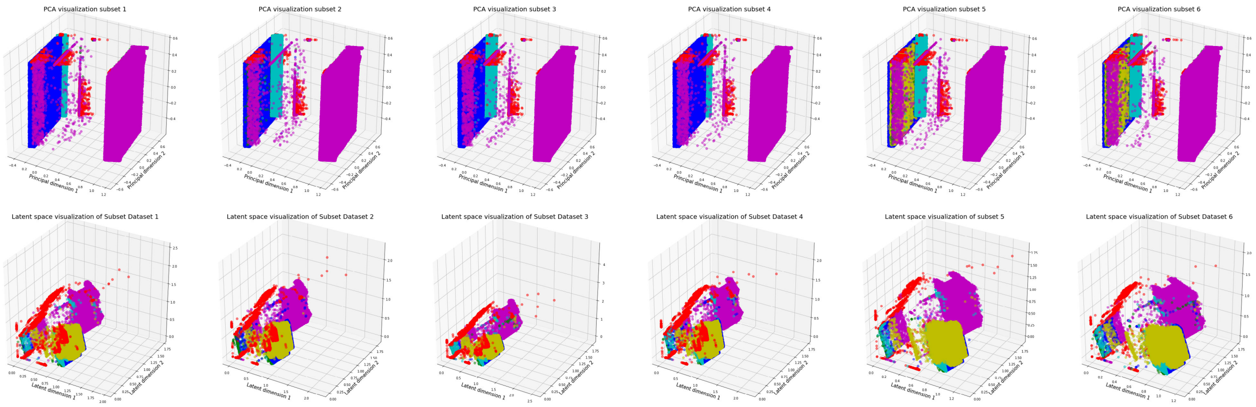
The plots in the top layer of Figure 5 show the PCA visualization of the distribution of data points of each test dataset in their raw form. As it is shown in the PCA visualization, there are more samples of DDoS attacks compared to the benign samples at each set - the similar pattern of data distribution we witnessed in the training dataset. The plots in the bottom layer of Figure 5 show the distribution of data points of each test set at the bottleneck layer of the trained AE which eventually becomes the inputs to the MLP model. As can be seen in the latent space visualization, there are distinct clusters around benign and each DDoS attack type. The size of the data points (i.e., number of features) that represent the benign and each DDoS attack type appear to be similar across all subdatasets. The detection and classification performance based on the performance metrics of Accuracy, Precision, Recall and F1-score on the five different DDoS attack types are shown in Table 7.

TABLE 7. Performance metrics on different DDoS attack types.

Testing Subdatasets	Attack types	performance metrics Accuracy	 Precision	 Recall	 F1
subdataset 1	BENIGN	99.95	92.12	88.19	90.11
	LDAP	99.21	93.91	98.30	96.05
	MSSQL	97.41	98.05	93.16	95.54
	NetBIOS	**99.96**	99.97	**99.82**	**99.90**
	SYN	99.94	**99.97**	99.76	99.86
	UDP	98.14	92.76	98.08	95.34
Subdataset 2	BENIGN	99.91	90.15	75.79	82.35
	LDAP	99.17	93.60	98.26	95.87
	MSSQL	98.50	98.06	96.89	97.47
	NetBIOS	**99.95**	99.92	**99.81**	**99.87**
	SYN	99.92	**99.93**	99.72	99.83
	UDP	99.24	98.11	97.96	98.03
Subdataset 3	BENIGN	99.90	72.34	95.74	82.41
	LDAP	99.18	93.71	98.30	95.95
	MSSQL	98.46	97.99	96.80	97.39
	NetBIOS	**99.85**	99.31	**99.90**	**99.60**
	SYN	99.76	**99.97**	98.93	99.45
	UDP	98.96	96.81	97.87	97.38
Subdataset 4	BENIGN	99.95	94.32	87.09	90.56
	LDAP	99.19	93.75	98.35	96.00
	MSSQL	97.71	98.05	94.18	96.08
	NetBIOS	**99.96**	99.92	**99.88**	**99.90**
	SYN	99.95	**99.98**	99.80	99.89
	UDP	98.46	94.28	97.99	96.10
Subdataset 5	BENIGN	99.95	95.26	85.62	90.18
	LDAP	99.18	93.79	98.24	95.96
	MSSQL	98.45	98.01	96.75	97.38
	NetBIOS	**99.96**	99.93	**99.87**	**99.90**
	SYN	99.94	**99.96**	99.79	99.87
	UDP	99.23	98.04	97.98	98.01
Subdataset 6	BENIGN	99.91	76.91	94.71	84.89
	LDAP	99.20	93.76	98.44	96.04
	MSSQL	97.63	97.99	93.98	95.94
	NetBIOS	**99.94**	99.84	**99.87**	**99.86**
	SYN	99.90	**99.99**	99.57	99.77
	UDP	98.37	94.09	97.75	95.87

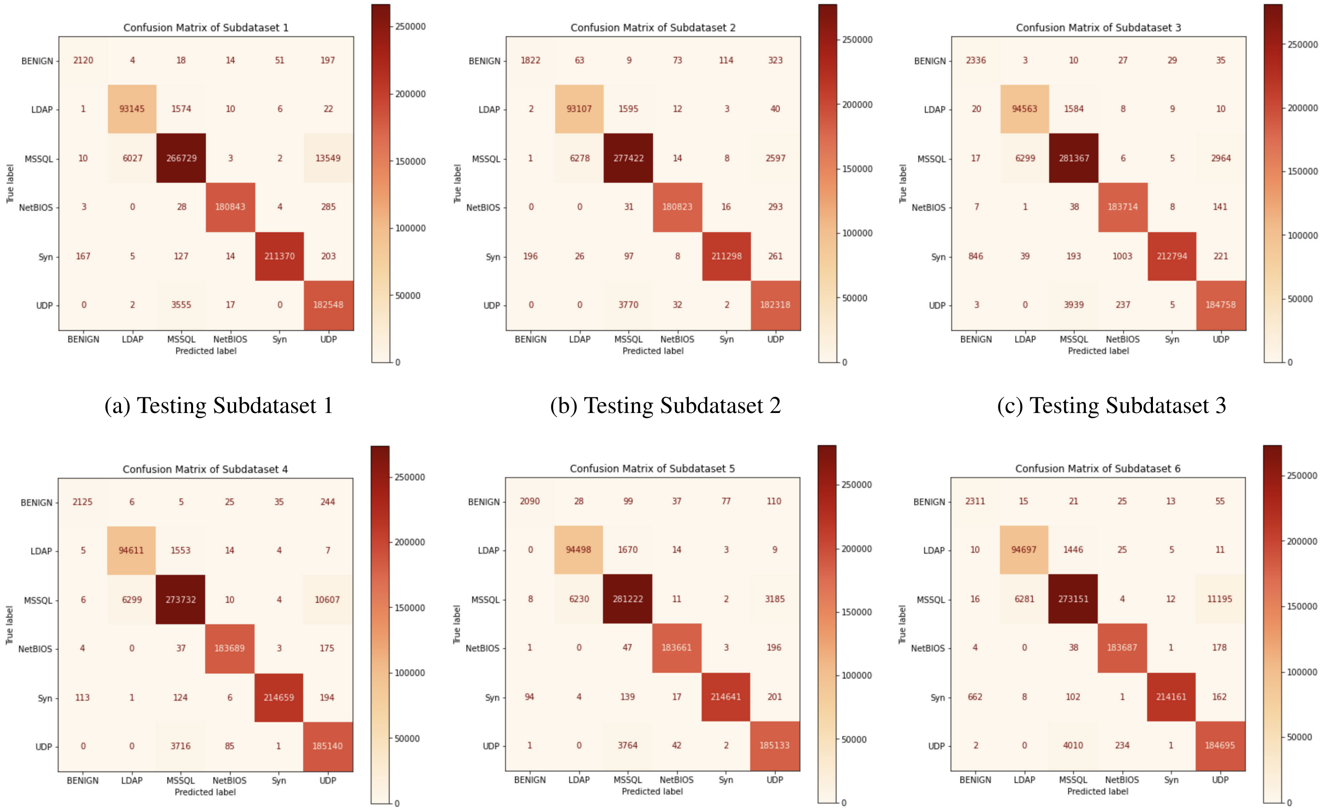
All different sub-datasets show very similar trends of the performance metrics which confirms that our proposed model does not overfit/underfit. Closely observing the performance metrics of each DDoS attack type, almost all DDoS attack types achieved above 97% classification accuracy. The NetBIOS attack type however showed the highest accuracy rate very close to almost 100%.

Figure 6 illustrates the exact number of records classified for different performance metrics for five DDoS attack types based on the confusion matrix. Similar to the results presented in Table 7, the DDoS attack type "SYN" has the most number of TPR where the number of FPR is almost negligent



(a) Subdataset 1 (b) Subdataset 2 (c) Subdataset 3 (d) Subdataset 4 (e) Subdataset 5 (f) Subdataset 6

FIGURE 5. The top layer shows PCA visualization of test datasets where axes indicate principal components of PCA embedding. The bottom layer shows latent space projection of test datasets through the bottleneck layer of trained AE where axes indicate latent space components of projection. (●: benign, ●: LDAP, ●: MSSQL, ● NetBios, ●: Syn, ●: UDP).



(a) Testing Subdataset 1 (b) Testing Subdataset 2 (c) Testing Subdataset 3

(d) Testing Subdataset 4 (e) Testing Subdataset 5 (f) Testing Subdataset 6

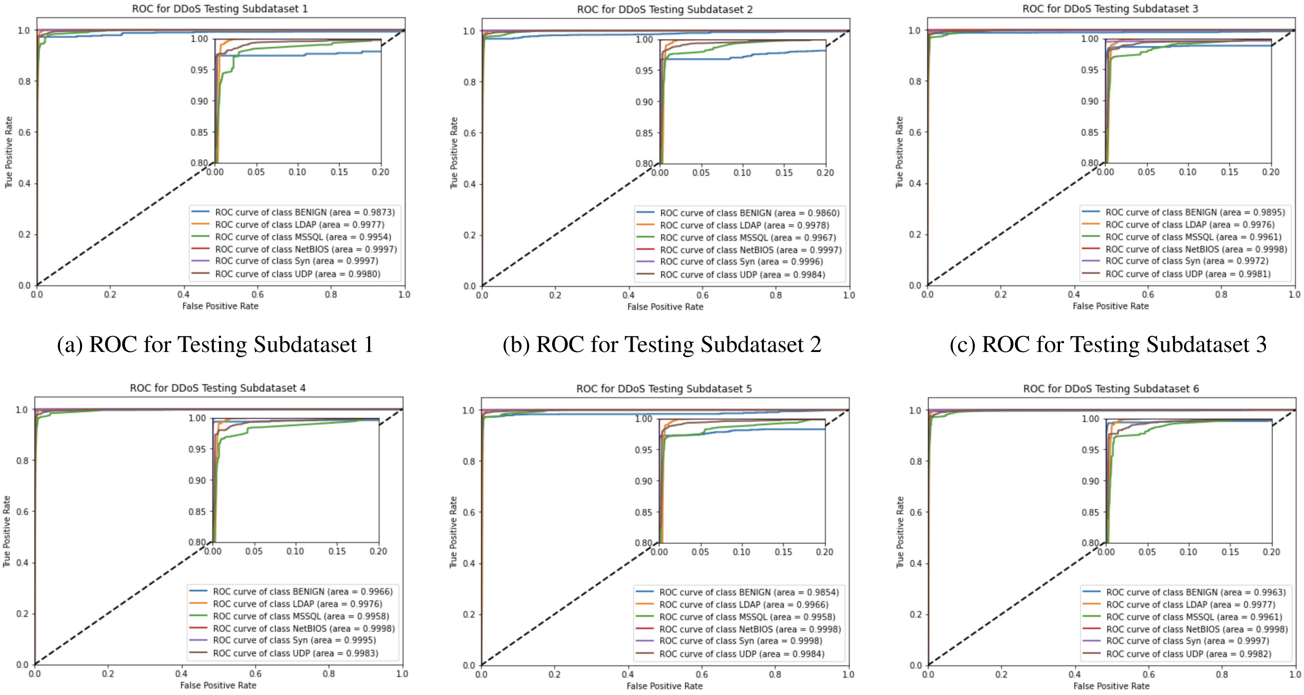
FIGURE 6. Performance of classification on different DDoS attack types based on confusion matrix.

(around a few hundred records misclassified). In comparison, the DDoS attack type "MSSQL" shows the worst performance where there is a large number of FPR that goes beyond thousand records.

The average performance metrics of Accuracy, Precision, Recall, and F1-score on different sub-datasets are shown in Table 8. The number of traffic samples contained in different sub-dataset differ from >960,000 (i.e., subdataset 1) to close to a million (i.e., subdataset 6). Regardless of the number of traffic samples, the different sub-dataset show a very similar pattern across all 6 subdatasets. The accuracy is in the range of 97% and 98% while the similar pattern in the Precision, Recall, and F1 scores are shown.

TABLE 8. Average performance on six subdatasets.

		Testing Subdataset					
	Attack types	Subdataset 1	Subdataset 2	Subdataset 3	Subdataset 4	Subdataset 5	Subdataset 6
No. of Instance	BENIGN	2,402	2,404	2,440	2,440	2,441	2,440
	LDAP	94,758	94,759	96,194	96,194	96,194	96,194
	MSSQL	286,320	286,320	290,658	290,658	290,658	290,659
	NetBIOS	181,163	181,163	183,909	183,908	183,908	183,908
	SYN	211,886	211,886	215,096	215,097	215,096	215,096
	UDP	186,122	186,122	188,942	188,942	188,942	188,942
Total performance mertrics		Acc = 97.31 Pre = 96.13 Recall = 96.22 F1 = 96.14	Acc = 98.35 Pre = 96.63 Recall = 94.74 F1 = 95.57	Acc = 98.19 Pre = 93.59 Recall = 97.91 F1 = 95.47	Acc = 97.62 Pre = 96.71 Recall = 96.21 F1 = 96.42	Acc = 98.36 Pre = 97.50 Recall = 96.37 F1 = 96.88	Acc = 97.49 Pre = 93.76 Recall = 97.39 F1 = 95.40



(a) ROC for Testing Subdataset 1

(b) ROC for Testing Subdataset 2

(c) ROC for Testing Subdataset 3

(d) ROC for Testing Subdataset 4

(e) ROC for Testing Subdataset 5

(f) ROC for Testing Subdataset 6

FIGURE 7. Performance based on AUC-ROC metric.

TABLE 9. Comparison to other similar methods.

Paper	Techniques	Acc	Precision	Recall	F1-score
Sharafaldin et al. [4]	Random Forrest	-	77	56	62
Rajagopal et al. [28]	Extended Decision Tree	97	99.0	97.0	97.8
Gohil et al. [8]	Extended Naive Bayes	96.25	96	96	96
Shieh et al. [15]	Bi-LSTM	98.18	97.93	99.84	-
De Assis et al. [29]	CNN	95.4	93.3	92.4	92.8
De Assis et al. [29]	MLP	92.5	84.4	94.2	89.0
Javaid et al. [20]	AE + Regression	88.39	85.44	95.95	90.4
Sadaf et al. [21]	AE + Isolation Forest	88.98	87.92	93.48	90.61
Can et al. [2]	FS + MLP	-	91.16	79.41	79.39
Our Proposal	AE-MLP	98.34	97.91	98.48	98.18

Figure 7 shows the AUC-ROC depicted on different DDoS attack types across all 6 subdatasets. The AUC-ROC value is more than 0.99 in all DDoS attack types in all subdatasets which confirms that our proposed AE-MLP model is highly effective in detecting and classifying different DDoS attack types with very high TPR while FPR stays very low.

C. COMPARISON TO OTHER SIMILAR METHODS

Table 9 shows the performance comparison of our proposed AE-MLP model with other similar methods both from shallow machine learning and deep learning-based neural network approaches. As the results show, our approach shows the best performance in terms of all aspects of performance metrics reaching the average of 98.34% accuracy while the precision, recall, and F1-score all remain very competitive at 97.91%, 98.48%, and 98.18% respectively. In general, shallow machine learning approaches do not perform as well as deep learning-based counterparts unless they are extended, for example, Data Jungle proposed by Rajagopal *et al.* [28] which combines several decision trees to achieve a higher accuracy rate. A deep learning-based approach using a standalone classifier, such as LSTM, CNN, and MLP tends to achieve more than 90% accuracy and demonstrate that they are suitable to provide an effective classifier to detect and classify different DDoS attack types. In the realm of hybrid approaches, AE combined with a shallow machine learning-based classifier such as using linear regression or an isolation forest tends to work less than when two deep learning models are combined like ours.

VI. CONCLUSION

In this study, we show that DDoS attacks can be detected and classified with high accuracy using the combination of deep learning-based techniques. Our proposed hybrid model AE-MLP consists of an Autoencoder (AE) and a Multi-layer Perceptron Network (MLP). The AE in our proposed model extracts the most important and relevant features to find malicious DDoS network payloads from a large-scale network traffic sample. The compressed and reduced features produced by the AE model is then fed to the MLP to effectively classify different DDoS attack types. This hybrid approach is not only effective in detecting DDoS attacks in a timely manner but is also effective in classifying what DDoS attack family the detected DDoS payload belongs to. Our proposed model can be an effective DDoS defense tool to detect a massively growing number of DDoS attacks in recent times. Our proposed model, comprehensively and extensively tested against many subsets of large DDoS attack samples, demonstrates high performances against many performance metrics such as Precision (97.91%), Recall (98.48%), F1-score (98.18%), and Accuracy (98.34%) which outperformed many other similar methods.

We have plans in place to apply different types of intrusion attack samples (e.g., Android-based malware samples [30] or ransomware [31], [32]) and other dataset samples from other applications (e.g., indoor air quality (IAQ) [33]–[37], medical annotations) to test the generalizability and practicability of our model.

REFERENCES

[1] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 39–53, Apr. 2004.

[2] D. C. Can, H. Q. Le, and Q. T. Ha, "Detection of distributed denial of service attacks using automatic feature selection with enhancement for imbalance dataset," in *Proc. ACIIDS*, 2021, pp. 386–398.

[3] X. Yuan, C. Li, and X. Li, "DeepDefense: Identifying DDoS attack via deep learning," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, May 2017, pp. 1–8.

[4] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2019, pp. 1–8.

[5] Y. B. Zikria, R. Ali, M. K. Afzal, and S. W. Kim, "Next-generation Internet of Things (IoT): Opportunities, challenges, and solutions," *Sensors*, vol. 21, no. 4, p. 1174, Feb. 2021.

[6] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, "Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset," *IEEE Access*, vol. 9, pp. 22351–22370, 2021.

[7] I. Ullah and Q. H. Mahmoud, "A technique for generating a botnet dataset for anomalous activity detection in IoT networks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2020, pp. 134–140.

[8] M. Gohil and S. Kumar, "Evaluation of classification algorithms for distributed denial of service attack detection," in *Proc. IEEE 3rd Int. Conf. Artif. Intell. Knowl. Eng. (AIKE)*, Dec. 2020, pp. 138–141.

[9] H. A. Alamri and V. Thayananthan, "Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against DDoS attacks," *IEEE Access*, vol. 8, pp. 194269–194288, 2020.

[10] T. T. Khoei, G. Aissou, W. C. Hu, and N. Kaabouch, "Ensemble learning methods for anomaly intrusion detection system in smart grid," in *Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT)*, May 2021, pp. 129–135.

[11] D. Parfenov, L. Kuznetsova, N. Yanishevskaya, I. Bolodurina, A. Zhigalov, and L. Legashev, "Research application of ensemble machine learning methods to the problem of multiclass classification of DDoS attacks identification," in *Proc. Int. Conf. Eng. Telecommun. (En&T)*, Nov. 2020, pp. 1–7.

[12] O. R. Sanchez, M. Repetto, A. Carrega, and R. Bolla, "Evaluating ML-based DDoS detection with grid search hyperparameter optimization," in *Proc. IEEE 7th Int. Conf. Netw. Softwarization (NetSoft)*, Jun. 2021, pp. 402–408.

[13] J. E. Varghese and B. Muniyal, "An efficient IDS framework for DDoS attacks in SDN environment," *IEEE Access*, vol. 9, pp. 69680–69699, 2021.

[14] C. F. T. Pontes, M. M. C. de Souza, J. J. C. Gondim, M. Bishop, and M. A. Marotta, "A new method for flow-based network intrusion detection using the inverse Potts model," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 2, pp. 1125–1136, Jun. 2021.

[15] C.-S. Shieh, W.-W. Lin, T.-T. Nguyen, C.-H. Chen, M.-F. Horng, and D. Miu, "Detection of unknown DDoS attacks with deep learning and Gaussian mixture model," *Appl. Sci.*, vol. 11, no. 11, p. 5213, Jun. 2021.

[16] S. U. Rehman, M. Khaliq, S. I. Imtiaz, A. Rasool, M. Shafiq, A. R. Javed, Z. Jalil, and A. K. Bashir, "DIDDOS: An approach for detection and identification of distributed denial of service (DDoS) cyberattacks using gated recurrent units (GRU)," *Future Gener. Comput. Syst.*, vol. 118, pp. 453–466, May 2021.

[17] M. Almiani, A. AbuGhazleh, Y. Jararweh, and A. Razaque, "DDoS detection in 5G-enabled IoT networks using deep Kalman backpropagation neural network," *Int. J. Mach. Learn. Cybern.*, vol. 12, pp. 1–13, Apr. 2021.

[18] P. S. Samom and A. Taggu, "Distributed denial of service (DDoS) attacks detection: A machine learning approach," in *Applied Soft Computing and Communication Networks*. Singapore: Springer, 2021, pp. 75–87.

[19] M. S. Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, "DDoSNet: A deep-learning model for detecting network attacks," in *Proc. IEEE 21st Int. Symp. World Wireless, Mobile Multimedia Networks (WoWMoM)*, Aug. 2020, pp. 391–396.

[20] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," *EAI Endorsed Trans. Secur. Saf.*, vol. 3, no. 9, pp. 1–6, 2016.

[21] K. Sadaf and J. Sultana, "Intrusion detection based on autoencoder and isolation forest in fog computing," *IEEE Access*, vol. 8, pp. 167059–167068, 2020.

[22] Y. N. Kunang, S. Nurmaini, D. Stiawan, A. Zarkasi, Firdaus, and Jasmir, "Automatic features extraction using autoencoder in intrusion detection system," in *Proc. Int. Conf. Electr. Eng. Comput. Sci. (ICECOS)*, Oct. 2018, pp. 219–224.

[23] J. Zhu, J. Jang-Jaccard, T. Liu, and J. Zhou, "Joint spectral clustering based on optimal graph and feature selection," *Neural Process. Lett.*, vol. 53, no. 1, pp. 257–273, Feb. 2021.

[24] T. T. Nguyen and V. J. Reddi, "Deep reinforcement learning for cyber security," 2019, *arXiv:1906.05799*.

[25] M. K. Alzaylaee, S. Y. Yerima, and S. Sezer, "DL-droid: Deep learning based Android malware detection using real devices," *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101663.

[26] T. Meng, X. Jing, Z. Yan, and W. Pedrycz, "A survey on machine learning for data fusion," *Inf. Fusion*, vol. 57, pp. 115–129, May 2020.

[27] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Appl. Sci.*, vol. 9, no. 20, p. 4396, Oct. 2019.

[28] S. Rajagopal, P. P. Kundapur, and H. K. S., "Towards effective network intrusion detection: From concept to creation on azure cloud," *IEEE Access*, vol. 9, pp. 19723–19742, 2021.

[29] M. V. O. de Assis, L. F. Carvalho, J. J. P. C. Rodrigues, J. Lloret, and M. L. Proença, Jr., "Near real-time security system applied to SDN environments in IoT networks using convolutional neural network," *Comput. Electr. Eng.*, vol. 86, Sep. 2020, Art. no. 106738.

[30] J. Zhu, J. Jang-Jaccard, and P. A. Watters, "Multi-loss Siamese neural network with batch normalization layer for malware detection," *IEEE Access*, vol. 8, pp. 171542–171550, 2020.

[31] T. R. McIntosh, J. Jang-Jaccard, and P. A. Watters, "Large scale behavioral analysis of ransomware attacks," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2018, pp. 217–229.

[32] T. McIntosh, J. Jang-Jaccard, P. Watters, and T. Susnjak, "The inadequacy of entropy-based ransomware detection," in *Proc. Int. Conf. Neural Inf. Process.* Springer, 2019, pp. 181–189.

[33] Y. Wei, J. Jang-Jaccard, F. Sabrina, and H. Alavizadeh, "Large-scale outlier detection for low-cost PM_{10} sensors," *IEEE Access*, vol. 8, pp. 229033–229042, 2020.

[34] Y. Wei, J. Jang-Jaccard, F. Sabrina, and T. McIntosh, "MSD-kmeans: A novel algorithm for efficient detection of global and local outliers," 2019, *arXiv:1910.06588*.

[35] R. Weyers, J. Jang-Jaccard, A. Moses, Y. Wang, M. Boulic, C. Chitty, R. Phipps, and C. and Cunningham, "Low-cost indoor air quality (IAQ) platform for healthier classrooms in New Zealand: Engineering issues," in *Proc. 4th Asia–Pacific World Congr. Comput. Sci. Eng. (APWC on CSE)*, Dec. 2017, pp. 208–215.

[36] Y. Wang, M. Boulic, R. Phipps, C. Chitty, A. Moses, R. Weyers, J. Jang-Jaccard, G. Olivares, A. Ponder-Sutton, and C. Cunningham, "Integrating open-source technologies to build a school indoor air quality monitoring box (SKOMOBO)," in *Proc. 4th Asia–Pacific World Congr. Comput. Sci. Eng. (APWC on CSE)*, Dec. 2017, pp. 216–223.

[37] Y. Wang, J. Jang-Jaccard, M. Boulic, R. Phipps, C. Chitty, R. Weyers, A. Moses, G. Olivares, A. Ponder-Sutton, and C. Cunningham, "Deployment issues for integrated open-source—Based indoor air quality school monitoring box (SKOMOBO)," in *Proc. IEEE Sensors Appl. Symp. (SAS)*, Mar. 2018, pp. 1–4.

YUANYUAN WEI received the master's degree in information technology from Massey University, Auckland, New Zealand, where she is currently pursuing the Ph.D. degree with the School of Natural and Computational Sciences. Her research interests include AI-powered anomaly detection, network intrusion detection, machine learning, and deep learning.

JULIAN JANG-JACCARD received the M.Sc. and Ph.D. degrees from The University of Sydney, Australia. She is currently an Associate Professor and the Head of the Cybersecurity Laboratory, Massey University, New Zealand. Prior to Massey, she was a Pioneering Member of the cybersecurity research at CSIRO, Australia's national science agency and innovation catalyst with a world-class reputation for their Research and Development, and developed the first set of practical cybersecurity solutions applied at Westpac and Telstra, Australia. She has published more than 70 papers in the leading conferences and journal venues, including IEEE and ACM. Her research interests include cybersecurity, intrusion detection, anomaly detection, artificial intelligence, data anonymization, and privacy-preservation techniques. She was a recipient of many multi-million dollar research awards both from Australian and New Zealand governments/industries, while collaborating with the top international ICT companies and universities around the world.

FARIZA SABRINA (Member, IEEE) received the Ph.D. degree in computer science and engineering from the University of New South Wales, Australia, and the Master of Engineering degree (by research) in electrical and information engineering from The University of Sydney, Australia. She has many years of research, teaching, and industrial experience in information and communication technologies. Currently, she is working as a Senior Lecturer and the Discipline Lead of networks and information security with the School of Engineering and Technology, Central Queensland University, Australia. Her current research interests include networking and information security, the Internet of Things (IoT), cybersecurity, blockchain, and artificial intelligence. She serves as a Technical Program Committee Member of various conferences. She is also a member of ACM and ACS.

AMARDEEP SINGH received the Ph.D. degree from Massey University, New Zealand, in 2021. His Ph.D. research focused on the use of machine learning and signal processing to minimize calibration time in an electroencephalography (EEG)-based brain–computer interface (BCI). Currently, he is working as a Research Officer at Massey University, where he is working on providing AI solutions for network intrusion/malware detection and classification.

WEN XU received the master's degree in information science from Massey University, Auckland, New Zealand. He is a Junior Research Officer with the School of Natural and Computational Sciences, Massey University. His current research interests include deep learning and AI-based network intrusion detection.

SEYIT CAMTEPE (Senior Member, IEEE) received the Ph.D. degree from Rensselaer Polytechnic Institute, in 2007. He was with the Technische Universität Berlin as a Senior Researcher and with QUT as a Lecturer. He is the Principal Research Scientist and the Team Leader at CSIRO's Data61. He was among the first to investigate the security of Android smartphones and inform society of the rising malware threat. His research interests include ML and cyber security, malware detection and prevention, smartphone security, applied and malicious cryptography, and CII security.

• • •