Noise-Robust Multilayer Perceptron Architecture for Distributed Denial of Service Attack Detection

João Paulo A. Maranhão, *Graduate Student Member, IEEE*, João Paulo C. L. da Costa, *Senior Member, IEEE*, Edison Pignaton de Freitas, *Member, IEEE*, Elnaz Javidi, *Graduate Student Member, IEEE*, and Rafael T. de Sousa, Jr., *Senior Member, IEEE*

***Abstract*— Distributed Denial of Service (DDoS) attacks are one of the most challenging security threats, since a single victim is attacked by several compromised malicious nodes. As a consequence, legitimate end users can be prevented to access network resources. This letter proposes a noise-robust multilayer perceptron (MLP) architecture for DDoS attack detection trained with corrupted data. In the proposed approach, the average value of the common features among dataset instances is iteratively filtered out by applying Higher Order Singular Value Decomposition (HOSVD) based techniques. The effectiveness of the proposed architecture is validated through comparison with state-of-the-art methods.**

***Index Terms*— Tensor decomposition, machine learning, supervised classification, neural networks.**

I. Introduction

DISTRIBUTED Denial of Service (DDoS) attacks are one of the most challenging threats to network security and present very sophisticated and damaging attacks [1]. DDoS attacks can be efficiently detected by using supervised machine learning (ML) techniques trained with large datasets such that malicious patterns present in the incoming network traffic can be detected with high reliability. Nonetheless, the classifier performance can be severely degraded if corrupted datasets are considered and, consequently, it is fundamental to develop DDoS attack detection models robust against noise present in data. For example, in [2], it was proposed a robust network intrusion detection system (IDS) in which redundant and irrelevant noisy informations are removed from the massive data through the combining of deep belief networks (DBN) and feature-weighted support vector machines (WSVM). In this work, noise refers to data corruptions as a consequence of false data injection attacks performed on publicly available datasets. For instance, Gaussian noise injection attacks are easy to implement in practice and aim to fool machine learning classifiers during the training and testing phases [3].

This letter proposes a novel multilayer perceptron (MLP) architecture for DDoS attack detection robust against data corruption. In our solution, the average value of the common features among dataset instances is dynamically filtered out via Higher Order Singular Value Decomposition (HOSVD) algorithm. The best MLP parameters used for dataset filtering are dynamically computed according to the errors between the expected and predicted classifications. Extensive experiments conducted on samples extracted from the CIC-IDS2017, CSE-CIC-IDS2018 and CIC-DDoS2019 datasets showed that the proposed solution outperforms state-of-the-art schemes in terms of accuracy, detection rate and false alarm rate.

Manuscript received October 3, 2020; accepted October 15, 2020. Date of publication October 19, 2020; date of current version February 11, 2021. This work was supported by CAPES PROAP/UnB/PPGEE, CNPq, FAPDF, ME, ABIN, AGU, CADE and MD/EB/DCT/CDS. The associate editor coordinating the review of this paper and approving it for publication was Y. Wu. *(Corresponding author: João Paulo A. Maranhão.)*

João Paulo A. Maranhão and Rafael T. de Sousa, Jr., are with the Department of Electrical Engineering, University of Brasília (UnB), Brasília 70910-900, Brazil (e-mail: joaopaulo.maranhao@ieee.org; desousa@unb.br).

João Paulo C. L. da Costa is with the Department of Electrical Engineering, University of Brasília (UnB), Brasília 70910-900, Brazil, and also with the Department 2-Campus Lippstadt, Hamm-Lippstadt University of Applied Sciences, 59063 Hamm, Germany (e-mail: joaopaulo.dacosta@ene.unb.br; joaopaulo.dacosta@hshl.de).

Edison Pignaton de Freitas is with the Informatics Institute, Federal University of Rio Grande do Sul (UFRGS), 90040-060 Porto Alegre, Brazil (e-mail: epfreitas@inf.ufrgs.br).

Elnaz Javidi is with the Department of Mechanical Engineering, University of Brasília (UnB), Brasília 70910-900, Brazil (e-mail: elnaz.javidi@gmail.com).

Digital Object Identifier 10.1109/LCOMM.2020.3032170

II. Data Model

In this letter, the dataset tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times \cdots \times N_R \times M}$ is defined as $\mathcal{X} = \mathcal{X}_0 + \mathcal{N}$, where N_r is the number of features along the r-th dimension, M is the number of instances, $\mathcal{X}_0 \in \mathbb{R}^{N_1 \times \cdots \times N_R \times M}$ is the noise-free dataset tensor and $\mathcal{N} \in \mathbb{R}^{N_1 \times \cdots \times N_R M}$ is the noise tensor. The r-th mode unfolding matrix of $\mathcal{X}$ is denoted by $[\mathcal{X}]_{(r)} \in \mathbb{R}^{N_r \times \prod_{j \neq r} N_j M}$. Moreover, $\mathbf{y} = [y_1, \ldots, y_M]^\mathsf{T} \in \mathbb{R}^M$ denotes the class label vector, where y_m is the class label of the m-th instance.

III. Proposed Noise-Robust MLP Architecture for DDoS Attack Detection

A. Proposed Feature Extraction Method

The concept of common and individual features is well known in image classification problems. For example, consider a 3D tensor, composed by multiple color matrices stacked along the 3rd dimension. Such matrices contain common features, which correspond to the base color matrices red, green and blue, as well as individual features, which provide discriminative information. The three base color matrices can be stacked in the 3rd dimension, constituting the common feature tensor. Thus, ML algorithms can benefit from exploiting such individual features for identifying each color matrix [4].

Applying such concepts to the DDoS attack detection problem, the following intuition can be drawn. Let us consider a 3D network traffic dataset composed by legitimate and DDoS attack sample matrices as its frontal slices. Although from different classes, these samples may present some common features, such as source and destination IPs. Therefore, the common features across all the samples can be removed and then the more discriminative individual features can be applied for classification.

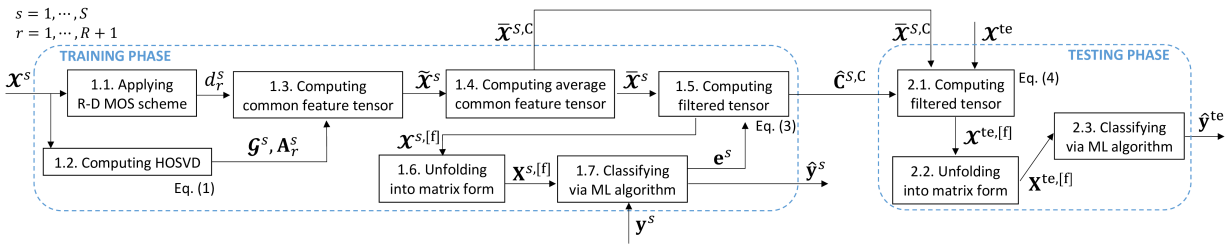


Fig. 1. Block diagram of the proposed feature extraction method applied on ML classification.

Next, the concepts relating the average common feature extraction and the noise-robustness of our proposed approach are presented. The tensors $\hat{\mathcal{X}} \in \mathbb{R}^{N_1 \times \cdots \times N_R \times d_{R+1}}$ and $\breve{\mathcal{X}} \in \mathbb{R}^{N_1 \times \cdots \times N_R \times M}$ are the common and individual feature tensors, respectively. The tensor $\hat{\mathcal{X}}$ can be obtained after applying some tensor decomposition technique on $\mathcal{X}$, such as the HOSVD, which is defined as $\mathcal{X} = \mathcal{G} \times_1 \mathbf{A}_1 \cdots \times_R \mathbf{A}_R \times_{R+1} \mathbf{A}_{R+1}$, where $\mathcal{G} \in \mathbb{R}^{d_1 \times \cdots \times d_{R+1}}$ is the truncated core tensor, $\mathbf{A}_r \in \mathbb{R}^{N_r \times d_r}$ for $r = 1, \ldots, R+1$ are the truncated singular matrices and $(d_1, \ldots, d_{R+1})$ is the multilinear rank of $\mathcal{X}$. The number of common features among dataset instances, given by d_{R+1}, can be obtained empirically. In addition, $\hat{\mathcal{X}} \in \mathbb{R}^{N_1 \times \cdots \times N_R \times d_{R+1}}$ is defined as the r-mode product between the core tensor $\mathcal{G}$ and the first R factor matrices, i.e., $\hat{\mathcal{X}} = \mathcal{G} \times_1 \mathbf{A}_1 \cdots \times_R \mathbf{A}_R$. Then, the individual feature tensor $\breve{\mathcal{X}}$ is computed by subtracting, from $\mathcal{X}$, the weighted common features, i.e., $\breve{\mathcal{X}}_{:,\ldots,m} = \mathcal{X}_{:,\ldots,m} - \sum_{k \in K_n} \alpha_k \cdot \hat{\mathcal{X}}_{:,\ldots,k}$ for $m = 1, \ldots, M$, where K_n is a subset of common features for $\mathcal{X}$ related to the values in the n-th row of $\mathbf{A}_r$ [4].

In this work, we propose an approach in which the DDoS detection performance is improved due to two factors: noise-robustness and dataset filtering. Noise-robustness is achieved through the HOSVD of the dataset tensor. In HOSVD, the singular value decomposition (SVD) is applied to each r-th unfolding matrix $[\mathcal{X}]_{(r)} = [\mathcal{X}_0]_{(r)} + [\mathcal{N}]_{(r)}$ and is given by $[\mathcal{X}]_{(r)} = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^{\mathsf{H}}$, where $\mathbf{U}_r \in \mathbb{R}^{N_r \times d_r}$ and $\mathbf{V}_r \in \mathbb{R}^{\prod_{j \neq r} N_j M \times d_r}$ are the left and right singular matrices, respectively, and $\mathbf{\Sigma}_r \in \mathbb{R}^{d_r \times d_r}$ is the singular values matrix.

On the other hand, after dataset filtering, ML algorithms can exploit the filtered features in order to identify each dataset instance. In this approach, instead of obtained empirically, the number of common features d_{R+1} can be approximated as the model order, which is a parameter computed from classic model order selection (MOS) techniques. Thus, a more discriminative information is present in each sample, which can be exploited by ML classifiers during the training phase.

Initially, three steps are necessary, namely, dataset splitting, dataset pre-processing and minibatch splitting. First, the dataset $\mathcal{X}$ is split into the training and testing tensors $\mathcal{X}^{\mathrm{tr}} \in \mathbb{R}^{N_1 \times \cdots \times N_R \times M^{\mathrm{tr}}}$ and $\mathcal{X}^{\mathrm{te}} \in \mathbb{R}^{N_1 \times \cdots \times N_R \times M^{\mathrm{te}}}$, where M^{tr} and M^{te} are the total number of training and testing instances, respectively, with $M = M^{\mathrm{tr}} + M^{\mathrm{te}}$. Next, a preprocessing step is applied on each tensor as well, including data cleansing, feature scaling and label encoding. Finally, $\mathcal{X}^{\mathrm{tr}}$ is split into S minibatches $\mathcal{X}^s \in \mathbb{R}^{N_1 \times \cdots \times N_R \times M^{\mathrm{mb}}}$ for $s = 1, \ldots, S$ containing M^{mb} instances each, i.e., $M^{\mathrm{tr}} = S \cdot M^{\mathrm{mb}}$. Additionally, if M^{tr} is not a multiple of M^{mb}, then random instances from $\mathcal{X}^{\mathrm{tr}}$ are added into the last minibatch such that the condition $M^{\mathrm{tr}} = S \cdot M^{\mathrm{mb}}$ is satisfied.

Fig. 1 depicts the block diagram of the proposed feature extraction method applied on the training and testing phases of ML classification.

1) Training Phase: First, given $\mathcal{X}^s$, the multilinear ranks d_r^s for $r = 1, \ldots, R+1$ and $s = 1, \ldots, S$ are estimated in Block 1.1 of Fig. 1 through multidimensional MOS schemes, such as the R-D Minimum Description Length (MDL) [5]. Further, Blocks 1.2 and 1.3 compute, respectively, the HOSVD of $\mathcal{X}^s \in \mathbb{R}^{N_1 \times \cdots \times N_R \times M^{\mathrm{mb}}}$ as well as the tensor $\hat{\mathcal{X}}^s \in \mathbb{R}^{N_1 \times \ldots N_R \times d_{R+1}}$, which contains the common features among the instances $\mathcal{X}^s_{:,\ldots,m} \in \mathbb{R}^{N_1 \times \cdots \times N_R}$ for $m = 1, \ldots, M^{\mathrm{mb}}$,

$$\begin{aligned} \mathcal{X}^s &= (\mathcal{G}^s \times_1 \mathbf{A}_1^s \cdots \times_R \mathbf{A}_R^s) \times_{R+1} \mathbf{A}_{R+1}^s \\ &= \hat{\mathcal{X}}^s \times_{R+1} \mathbf{A}_{R+1}^s, \end{aligned} \tag{1}$$

where $\mathcal{G}^s \in \mathbb{R}^{d_1^s \times \cdots \times d_{R+1}^s}$ is the core tensor, $\mathbf{A}_r^s \in \mathbb{R}^{N_r \times d_r^s}$ for $r = 1, \ldots, R+1$ are the singular matrices, and $(d_1^s, \ldots, d_{R+1}^s)$ corresponds to the multilinear rank of $\mathcal{X}^s$.

Next, $\bar{\mathcal{X}}^s \in \mathbb{R}^{N_1 \times \cdots \times N_R}$ is obtained in Block 1.4 of Fig. 1 and corresponds to $\hat{\mathcal{X}}^s$ averaged along the $(R+1)$-th dimension, i.e., $\bar{\mathcal{X}}^s = (1/d_{R+1}^s) \sum_{x=1}^{d_{R+1}^s} \hat{\mathcal{X}}^s_{:,\ldots,x}$. Then, the filtered tensor $\mathcal{X}^{s,[\mathrm{f}]}_{:,\ldots,m}$ for $m = 1, \ldots, M^{\mathrm{mb}}$ can be computed in Block 1.5 as follows

$$\mathcal{X}^{s,[\mathrm{f}]}_{:,\ldots,m} = \mathcal{X}^s_{:,\ldots,m} - \mathcal{C}^s \odot \bar{\mathcal{X}}^s, \quad \text{for} \quad m = 1, \ldots, M^{\mathrm{mb}}, \tag{2}$$

where $\mathcal{C}^s \in \mathbb{R}^{N_1 \times \cdots \times N_R}$ is the weight tensor.

Alternatively, M^{mb} identical copies of $\mathcal{C}^s$ and $\bar{\mathcal{X}}^s$ can be stacked along the $(R+1)$-th dimension, generating the concatenated tensors $\mathcal{C}^{s,\mathsf{C}} \in \mathbb{R}^{N_1 \times \cdots \times N_R \times M^{\mathrm{mb}}}$ and $\bar{\mathcal{X}}^{s,\mathsf{C}} \in \mathbb{R}^{N_1 \times \cdots \times N_R \times M^{\mathrm{mb}}}$. Consequently, (2) can be rewritten as

$$\mathcal{X}^{s,[\mathrm{f}]} = \mathcal{X}^s - \mathcal{C}^{s,\mathsf{C}} \odot \bar{\mathcal{X}}^{s,\mathsf{C}}. \tag{3}$$

Moreover, in Block 1.6 of Fig. 1, $\mathcal{X}^{s,[\mathrm{f}]}$ is unfolded into the matrix $\mathbf{X}^{s,[\mathrm{f}]} \in \mathbb{R}^{M^{\mathrm{mb}} \times N}$ and then forwarded for machine learning classification in Block 1.7. Finally, the error $\mathbf{e}^s$ between the expected and predicted class label vectors $\mathbf{y}^s$ and $\hat{\mathbf{y}}^s$, respectively, is computed and sent to Block 1.5 in order to update $\mathcal{C}^s$ before the next minibatch training.

2) Testing Phase: The testing phase is composed by three steps, as depicted in Boxes 2.1 to 2.3 of Fig. 1. Instead of applying a scheme similar to the one described for the training set, we simply apply a transformation to each testing instance by using information extracted from the training phase. Since testing and training instances are extracted from the same dataset, they have similarities and, consequently, we consider

that the testing dataset presents the same average common features of the training dataset.

First, the filtered testing dataset $\mathcal{X}^{\mathrm{te,[f]}}$ is computed in Box 2.1 by subtracting, from $\mathcal{X}^{\mathrm{te}}$, the weighted common feature tensor obtained from the training phase, i.e.,

$$\mathcal{X}^{\mathrm{te,[f]}} = \mathcal{X}^{\mathrm{te}} - \hat{\mathcal{C}}^{S,\mathsf{C}} \odot \bar{\mathcal{X}}^{S,\mathsf{C}}, \tag{4}$$

where $\hat{\mathcal{C}}^{S,\mathsf{C}}$ and $\bar{\mathcal{X}}^{S,\mathsf{C}}$ are, respectively, the weight tensor and the average common feature tensor, both obtained after the training of the S-th minibatch. Next, $\mathcal{X}^{\mathrm{te,[f]}}$ is unfolded into the matrix $\mathbf{X}^{\mathrm{te,[f]}} \in \mathbb{R}^{M^{\mathrm{te}} \times N}$, as shown in Box 2.2 of Fig. 1. Finally, $\mathbf{X}^{\mathrm{te,[f]}}$ is applied on the trained machine learning classifier in Box 2.3 for testing purposes, generating the predicted class label vector $\hat{\mathbf{y}}^{\mathrm{te}}$.

B. Proposed MLP Architecture

The proposed feature extraction method introduced in Subsection III-A is a pre-processing algorithm where denoising and filtering are performed over the dataset before ML classification. In this subsection, the proposed technique is directly applied on the MLP layers, and not as a pre-processing scheme. The training and testing phases of the proposed MLP architecture are described as follows.

1) Training Phase: Multilayer perceptrons are fully connected feedforward neural networks where the output of one layer is the input of the subsequent layer. If the MLP presents J layers, the output vector $\mathbf{z}^{[j]} \in \mathbb{R}^{N^{[j]}}$ of the j-th layer after receiving the input vector $\mathbf{z}^{[j-1]} \in \mathbb{R}^{N^{[j-1]}}$ from the $(j-1)$-th layer is given by

$$\mathbf{z}^{[j]} = f^{[j]}(\mathbf{W}^{[j,j-1]} \cdot \mathbf{z}^{[j-1]} + \mathbf{b}^{[j]}), \tag{5}$$

where $\mathbf{W}^{[j,j-1]} \in \mathbb{R}^{N^{[j]} \times N^{[j-1]}}$ is the weight matrix, $\mathbf{b}^{[j]} \in \mathbb{R}^{N^{[j]}}$ is the bias vector, $f^{[j]}(\cdot)$ is the activation function and $N^{[j]}$ is the number of neurons for $j = 1, \ldots, J$.

Since MLPs get one-dimensional input data, the input dataset tensor must be matricized such that each instance can be directly forwarded to the neural network. In this sense, first we compute the $(R+1)$-th mode unfolding of (3) as follows

$$[\mathcal{X}^{s,[\mathrm{f}]}]_{(R+1)} = [\mathcal{X}^{s}]_{(R+1)} - [\mathcal{C}^{s,\mathsf{C}}]_{(R+1)} \odot [\bar{\mathcal{X}}^{s,\mathsf{C}}]_{(R+1)}. \tag{6}$$

The weight tensor $\mathcal{C}^{s,\mathsf{C}}$ is composed by M^{mb} identical tensors $\mathcal{C}^{s}$ stacked along the $(R+1)$-th dimension. Consequently, the unfolded matrix $[\mathcal{C}^{s,\mathsf{C}}]_{(R+1)} \in \mathbb{R}^{M^{\mathrm{mb}} \times N}$ presents M^{mb} identical row vectors $\mathbf{c}^{s} = [c_1^s, \ldots, c_N^s] = \mathrm{vec}\{\mathcal{C}^{s}_{:,\ldots,m}\}$ for $m = 1, \ldots, M^{\mathrm{mb}}$. Alternatively, $\mathbf{c}^{s}$ can also be written as a diagonal matrix $\mathbf{C}^{s,\mathrm{diag}} = \mathrm{diag}\{[c_1^s, \ldots, c_N^s]\} \in \mathbb{R}^{N \times N}$.

Similarly, $[\bar{\mathcal{X}}^{s,\mathsf{C}}]_{(R+1)}$ is composed by M^{mb} identical row vectors $\bar{\mathbf{x}}^{s} = [\bar{x}_1^s, \ldots, \bar{x}_N^s] = \mathrm{vec}\{\bar{\mathcal{X}}^{s}_{:,\ldots,m}\}$ for any $m = 1, \ldots, M^{\mathrm{mb}}$. If the $(R+1)$-th mode unfolding matrices in (6) are represented as a function of its rows and replace the Hadamard product by the dot product, such equation can be rewritten as

$$[\mathcal{X}^{s,[\mathrm{f}]}]^{\mathsf{T}}_{(R+1)m,:} = -\mathbf{C}^{s,\mathrm{diag}} \cdot \bar{\mathbf{x}}^{s^{\mathsf{T}}} + [\mathcal{X}^{s}]^{\mathsf{T}}_{(R+1)m,:}. \tag{7}$$

Note the similarity between (7) and the argument of the activation function $f^{[j]}(\cdot)$ in (5). In this sense, a new input layer is included on the conventional MLP architecture where the computation described in (7) can be directly performed. Fig. 2 illustrates the proposed MLP architecture for DDoS attack detection with the new input layer, labeled as "layer 0", in red color, placed at the left of the conventional input layer, referred as "layer 1". Thus, by comparing (5) and (7), we set $N^{[0]} = N$, $\mathbf{W}^{[1,0]} = -\mathbf{C}^{s,\mathrm{diag}}$, $\mathbf{z}^{[0]} = \bar{\mathbf{x}}^{s^{\mathsf{T}}}$ and $\mathbf{b}^{[0]} = [\mathcal{X}^{s}]^{\mathsf{T}}_{(R+1)m,:}$ in the proposed MLP.

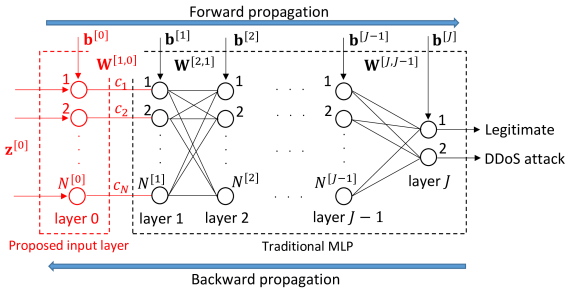


Fig. 2. Proposed MLP architecture.

2) Testing Phase: In this phase, each testing instance is feedforwarded through the trained MLP for classification. Consequently, the filtering operation on the testing vectors, for $m = 1, \ldots, M^{\mathrm{te}}$, is given by

$$[\mathcal{X}^{\mathrm{te,[f]}}]^{\mathsf{T}}_{(R+1):,m} = -\hat{\mathbf{C}}^{S,\mathrm{diag}} \cdot \bar{\mathbf{x}}^{S^{\mathsf{T}}} + [\mathcal{X}^{\mathrm{te}}]^{\mathsf{T}}_{(R+1):,m}, \tag{8}$$

where $\hat{\mathbf{C}}^{S,\mathrm{diag}}$ is the weight diagonal matrix and $\bar{\mathbf{x}}^{S}$ is the average common feature vector, both obtained after the training of the S-th minibatch.

IV. Experiments and Results

A. Results

This subsection presents the performance evaluation of the proposed MLP architecture through numerical simulations. Accuracy (Acc), detection rate (DR) and false alarm rate (FAR) are the assessed metrics. Detailed explanation about these metrics can be found in [6].

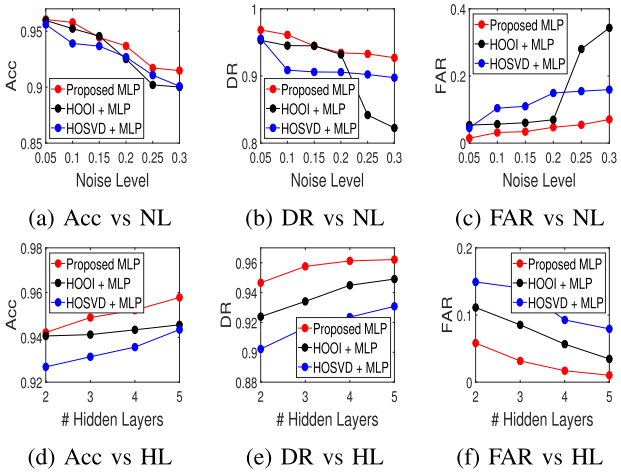
We customized a single dataset composed by legitimate and DDoS attack samples extracted from the CIC-DDoS2019, CIC-IDS2018 and CIC-IDS2017 benchmark datasets. Initially, the dataset is split into training and testing sets. Next, a 3-fold cross validation is performed on the training dataset such that each fold contains samples from a given benchmark dataset. At each iteration, the classifier is trained on samples from two benchmark datasets and validated on the third one. Finally, the trained classifier is evaluated on the testing dataset. The customized dataset is composed by $N = 64$ features and $M = 40000$ instances, of which 32000 are legitimate and 8000 are DDoS attacks. The dataset is folded as a three-dimensional tensor with size $N_1 \times N_2 \times M$, where $N_1 = N_2 = 8$.

In order to simulate false data injection, noise is added to each dataset prior to the pre-processing phase. As pointed out in [7], Gaussian noise is easy to implement and more difficult to be detected in practice. Thus, $x\%$ of the instances of each feature $\mathbf{X}_{:,n}$ for $n = 1, \ldots, N$ are corrupted with Gaussian noise with mean zero and standard deviation $(\max(\mathbf{X}_{:,n}) - \min(\mathbf{X}_{:,n}))/5$. A total of 100 different experiments were

TABLE I

PERFORMANCE EVALUATION FOR DIFFERENT RANKS

Metric	Multilinear rank (d_1, d_2, d_3)					
	Varying d_3			Varying d_1,d_2		
	(6,6,30)	(6,6,40)	(6,6,50)	(6,6,60)	(7,7,60)	(8,8,60)
Accuracy	0.9877	0.9876	0.9870	0.9866	0.9863	0.9860
Training time (s)	19.09	19.42	19.58	19.63	19.73	19.75



(a) Acc vs NL (b) DR vs NL (c) FAR vs NL

(d) Acc vs HL (e) DR vs HL (f) FAR vs HL

Fig. 3. Acc, DR and FAR as a function of the noise level and number of hidden layers.

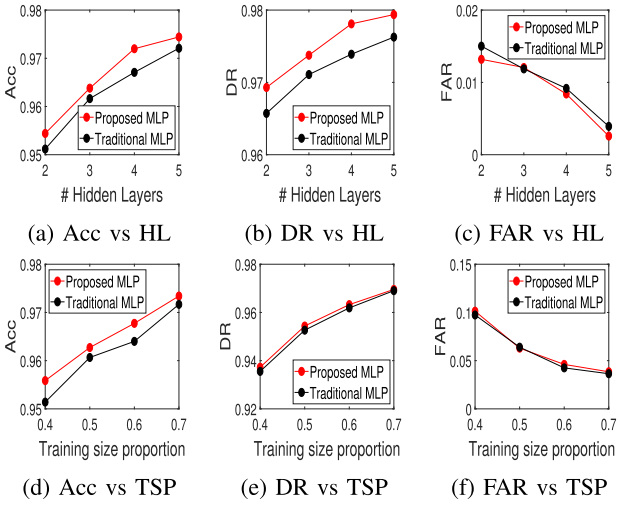
simulated, with a MLP containing number of hidden layers varying from 2 to 5 and noise level between 5% and 30%. Additionally, the training dataset size ranges from 40% to 70% of all available instances. The tensor multilinear rank is estimated by applying the R-D MDL scheme and each experiment was trained for 100 epochs.

The benefits of HOSVD on the overall performance of our proposed MLP are shown in Table I, which presents the accuracy and training times for different multilinear ranks. According to the obtained results, the proposed MLP achieves better performance for lower values of d_1, d_2 and d_3, which illustrates the efficiency of HOSVD for dataset denoising.

Next, the proposed approach is compared with conventional MLPs in which state-of-the-art low-rank approximation techniques are previously applied to the dataset, namely, HOSVD [8] and Higher Order Orthogonal Iteration (HOOI) [9]. Fig. 3 shows the Acc, DR and FAR as a function of the noise level (NL) and the number of hidden layers (HL). Moreover, in Fig. 3d to 3f, the NL fixed in 10%. Note that the proposed MLP outperforms its competitor methods, especially under high noise levels and larger number of hidden layers.

Fig. 4 depicts the values of Acc, DR and FAR obtained for the noise-free case. The proposed scheme is compared with a conventional MLP, with no denoising technique, for different HL and TSP. Note that, considering Acc and DR, the proposed scheme outperforms its competitor for all HL and TSP ranges, despite presenting a worse performance of FAR for some values of HL and TSP.

Following, the proposed MLP is assessed for detecting real time DDoS attacks. Three IDSs trained and tested under noise levels between 10% and 30% are compared: the proposed scheme and the HOSVD and HOOI based MLPs. To simulate a small scale DDoS attack, a virtual network was implemented, as depicted in Fig. 5. DDoS traffic included TCP, UDP and



(a) Acc vs HL (b) DR vs HL (c) FAR vs HL

(d) Acc vs TSP (e) DR vs TSP (f) FAR vs TSP

Fig. 4. Acc, DR and FAR as a function of the number of hidden layers and training size proportion under noise-free conditions.

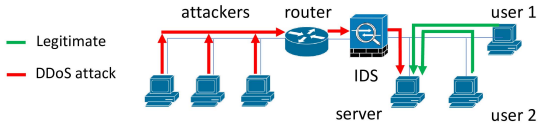


Fig. 5. Network topology for simulating real time attacks.

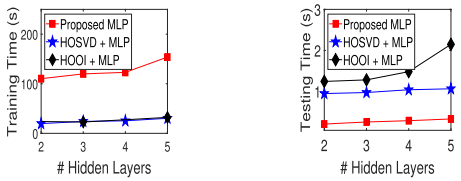
TABLE II

PERFORMANCE EVALUATION UNDER REAL TIME ATTACKS

Model	10%		20%		30%	
	DR	FAR	DR	FAR	DR	FAR
Proposed MLP	0.9967	0.0001	0.9661	0.0447	0.8373	0.1792
HOSVD + MLP	0.9413	0.1173	0.8485	0.1256	0.7697	0.3894
HOOI + MLP	0.9999	0.0000	0.8850	0.0659	0.7736	0.3088

HTTP GET flooding attacks generated by three attackers, whereas the victim is a web server. The attacks were launched during a period of 60 minutes by using the Low Orbit Ion Cannon (LOIC) tool. Additionally, legitimate traffic was generated by two users accessing the web server, and the IDS is positioned between the router and the victim. The network traffic captured by the IDS is converted into CSV files for further processing. Table II shows the values of DR and FAR obtained for real time detection. Note that the proposed scheme outperforms the compared approaches when NL is higher than 20%, presenting considerable detection results.

Further, Fig. 6 presents the mean training and testing times (in seconds), considering different numbers of hidden layers. Three techniques are compared: the proposed scheme, as well as the HOSVD and HOOI based MLPs. Since denoising and filtering are performed through the MLP layers in our proposed scheme, its total time correspond to the MLP processing time. On the other hand, since HOSVD and HOOI are preprocessing steps in their respective MLPs, the total time correspond to the sum between the corresponding low-rank approximation technique time and the MLP processing time. The training times refer to a period of 100 epochs and NL is fixed in 10%. Note that, from Fig. 6a, the proposed technique is more computationally expensive than the competing approaches, showing higher training times, which reflects the trade-off to achieve a more accurate detection. However, this is compensated considering the testing times, as shown in Fig. 6b.



(a) Training time (s) vs HL (b) Testing time (s) vs HL

Fig. 6. Mean training and testing times (in seconds).

TABLE III

PERFORMANCE COMPARISON WITH RELATED WORKS CONSIDERING: (1) NSL-KDD, AND (2) CIC-IDS2017

Dataset	Work	Algorithm	Acc	DR	FAR
(1)	Proposed scheme (NL = 30%)	MLP	0.9957	0.9953	0.0079
	Wu *et al.* [2]	DBN+SVM	0.9303	0.8724	0.0211
(2)	Proposed scheme (NL = 0%)	MLP	0.9895	0.9831	0.0015
	Roopak *et al.* [11]	MLP	0.8634	0.8625	N/A
	Doriguzzi-Corin *et al.* [10]	CNN	0.9967	0.9994	0.0059
	Lima Filho *et al.* [6]	RF	N/A	0.8000	0.0020

Finally, Table III illustrates the comparison with related works. To the best of our knowledge, only Wu *et al.* [2] proposed a noise-robust solution applied to network intrusion detection in which the NSL-KDD dataset was considered for validation. Since [2] did not inform the noise level, the proposed MLP was simulated with the worst case of NL $= 30\%$. Note that our proposed scheme presents a high noise-robustness, with Acc $= 99.57\%$, DR $= 99.53\%$ and FAR $= 0.79\%$. Additionally, since CIC-IDS2017 has been extensively applied for noise-free IDS validation by several works in the literature, the performance evaluation considering such dataset is also included. In this case, NL $= 0$ was adopted in our approach. Despite the detection performance of the proposed MLP is not the best when considering CIC-IDS2017, it still presents outstanding results, with Acc $= 98.95\%$, DR $= 98.31\%$ and FAR $= 0.15\%$. Note that our scheme is outperformed by [10], in which a convolutional neural network (CNN) based IDS is proposed. In [10], DDoS attacks and legitimate traffic patterns are learnt by CNN through convolutional filters sliding over packet flow inputs to identify anomalous characteristics, which may explain its higher Acc and DR results compared to our MLP based solution.

B. Discussion

From the results shown in the previous subsection, we observe that, when noisy datasets are considered, the proposed approach is more efficient for detecting DDoS attacks compared to HOSVD and HOOI based MLPs in most of the simulations. However, the accuracy of our scheme is matched by HOOI based MLP in scenarios with low NL. In this case, the more accurate core tensor and singular matrices generated through orthogonal iterations in HOOI lead to a better dataset denoising. Moreover, the HOSVD based MLP presents the worst performance, since a single tensor decomposition is performed on the whole dataset during the preprocessing phase. Additionally, under noise-free conditions, the proposed approach outperforms the conventional MLP in terms of Acc and DR for multiple HL and TSP. However, our scheme is more prone to false positives and, hence, legitimate traffic is wrongly classified as malicious activity. Therefore, the proposed MLP presents a higher noise-robustness and efficiency due to two factors: the noise attenuation provided by HOSVD and the more discriminative individual information resulting from dataset filtering. Furthermore, a drawback of our approach is its higher training time, caused by multiple HOSVD performed in training data batches. On the other hand, the proposed scheme shows lower testing times, since low cost computations are performed during the testing phase, in contrast to the tensor decompositions executed in HOSVD and HOOI. Finally, our proposed approach is more accurate for detecting real time DDoS attacks in comparison with HOOI and HOSVD based MLPs when NL $\geq 20\%$. Such results reflect the superiority of our scheme due to a more efficient dataset filtering during the training phase. Nonetheless, the best performance for NL $= 10\%$ was shown by HOOI based MLP, in which orthogonal iterations provided a more accurate separation between signal and noise subspaces and, consequently, better classifications.

V. CONCLUSION

This letter reports the proposal of a noise-robust MLP architecture for DDoS attack detection trained with corrupted datasets. The presented solution dynamically filter out the average value of the common features among dataset instances, improving the model performance. The proposed approach outperforms the competing techniques in almost all performance metrics, showing outstanding values of accuracy, detection rate and false alarm rate.

REFERENCES

[1] X. Liang and T. Znati, "An empirical study of intelligent approaches to DDoS detection in large scale networks," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2019, pp. 821–827.

[2] Y. Wu, W. W. Lee, Z. Xu, and M. Ni, "Large-scale and robust intrusion detection model combining improved deep belief network with feature-weighted SVM," *IEEE Access*, vol. 8, pp. 98600–98611, 2020.

[3] F. Li and Y. Tang, "False data injection attack for cyber-physical systems with resource constraint," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 729–738, Feb. 2020.

[4] I. Kisil, G. G. Calvi, and D. P. Mandic, "Tensor valued common and individual feature extraction: Multi-dimensional perspective," 2017, *arXiv:1711.00487*. [Online]. Available: http://arxiv.org/abs/1711.00487

[5] J. P. C. L. da Costa, F. Roemer, M. Haardt, and R. T. de Sousa, "Multi-dimensional model order selection," *EURASIP J. Adv. Signal Process.*, vol. 2011, no. 1, pp. 1–13, Dec. 2011.

[6] F. S. D. L. Filho, F. A. F. Silveira, A. de Medeiros Brito Junior, G. Vargas-Solar, and L. F. Silveira, "Smart detection: An online approach for DoS/DDoS attack detection using machine learning," *Secur. Commun. Netw.*, vol. 2019, pp. 1–15, Oct. 2019.

[7] Y. Li, Z. Pan, D. Du, and R. Li, "Adaptive thresholding HOSVD with rearrangement of tensors for image denoising," *Multimedia Tools Appl.*, vol. 79, nos. 27–28, pp. 19575–19593, Jul. 2020.

[8] A. Rajwade, A. Rangarajan, and A. Banerjee, "Image denoising using the higher order singular value decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 849–862, Apr. 2013.

[9] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-(R_1,R_2,...,R_n) approximation of higher-order tensors," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1324–1342, 2000.

[10] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martinez-del-Rincon, and D. Siracusa, "Lucid: A practical, lightweight deep learning solution for DDoS attack detection," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 2, pp. 876–889, Jun. 2020.

[11] M. Roopak, G. Y. Tian, and J. Chambers, "Deep learning models for cyber security in IoT networks," in *Proc. IEEE 9th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2019, pp. 0452–0457.